Operations Research

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Online Allocation and Pricing: Constant Regret via Bellman Inequalities

Alberto Vera, Siddhartha Banerjee, Itai Gurvich

To cite this article:
Alberto Vera, Siddhartha Banerjee, Itai Gurvich (2021) Online Allocation and Pricing: Constant Regret via Bellman Inequalities. Operations Research

Published online in Articles in Advance 26 Mar 2021

. https://doi.org/10.1287/opre.2020.2061

http://pubsonline.informs.org/journal/opre

OPERATIONS RESEARCH
Articles in Advance, pp. 1–20
ISSN 0030-364X (print), ISSN 1526-5463 (online)

Crosscutting Areas

Online Allocation and Pricing: Constant Regret via Bellman Inequalities

Alberto Vera,[a] Siddhartha Banerjee,[a] Itai Gurvich[b]

[a] School of Operations Research and Information Engineering, Cornell University, Ithaca, New York 14853; [b] Kellogg School of Management, Northwestern University, Evanston, Illinois 60208

Contact: aav39@cornell.edu, https://orcid.org/0000-0003-0569-8623 (AV); sbanerjee@cornell.edu, https://orcid.org/0000-0002-8954-4578 (SB); gurvich@cornell.edu, https://orcid.org/0000-0001-9746-7755 (IG)

Received: June 17, 2019
Revised: May 30, 2020
Accepted: June 11, 2020
Published Online in Articles in Advance: March 26, 2021

Subject classifications: dynamic programming/optimal control; applications
Area of Review: Revenue Management and Market Analytics

https://doi.org/10.1287/opre.2020.2061

Abstract. We develop a framework for designing simple and efficient policies for a family of online allocation and pricing problems that includes online packing, budget-constrained probing, dynamic pricing, and online contextual bandits with knapsacks. In each case, we evaluate the performance of our policies in terms of their regret (i.e., additive gap) relative to an offline controller that is endowed with more information than the online controller. Our framework is based on Bellman inequalities, which decompose the loss of an algorithm into two distinct sources of error: (1) arising from computational tractability issues, and (2) arising from estimation/prediction of random trajectories. Balancing these errors guides the choice of benchmarks, and leads to policies that are both tractable and have strong performance guarantees. In particular, in all our examples, we demonstrate constant-regret policies that only require resolving a linear program in each period, followed by a simple greedy action-selection rule; thus, our policies are practical as well as provably near optimal.

Funding: This work was supported by the U.S. Department of Defense [Grants STTR A18B-T007 and W911NF-20-C-0008], the National Science Foundation [Grants CNS-1955997, DMS-1839346, and ECCS-1847393], and the Army Research Laboratory [Grant W911NF-17-1-0094].
Supplemental Material: The online appendices are available at https://doi.org/10.1287/opre.2020.2061.

Keywords: stochastic optimization • approximate dynamic programming • online resource allocation • dynamic pricing • online packing • network revenue management

1. Introduction

Online decision making under uncertainty is widely studied across a variety of fields, including operations research, control, and computer science. A canonical framework for such problems is that of Markov decision processes (MDPs), with associated use of stochastic dynamic programming for designing policies. In complex settings, however, such approaches suffer from the known curse of dimensionality; moreover, they also fail to provide insights into structural properties of the problem: the performance of heuristics, dependence on distributional information, and so on.

These challenges have inspired an alternate approach to designing approximate policies for MDPs based on the use of benchmarks—proxies for the value function that provide bounds for the optimal policy and guide the design of heuristics. The performance of any policy can be quantified by its additive loss, or regret, relative to any such benchmark; this consequently also bounds the additive optimality gap, that is, performance against the optimal policy.

In this work, we develop new policies for online resource-allocation problems: settings where a finite set of resources is dynamically allocated to arriving requests, with associated constraints and rewards/costs. Our baseline problem is the online stochastic knapsack problem (henceforth OnlineKnapsack): a controller has initial inventory B, and requests arrive sequentially over horizon T. Each request has a random type corresponding to a resource requirement-reward pair. Requests are generated from a known stochastic process, and are revealed upon arrival; the controller must then decide whether to accept/reject each request in order to maximize rewards while satisfying budget constraints. We then consider three variants of this basic setting: (1) online probing, (2) dynamic pricing, and (3) contextual bandits with knapsacks. These are widely studied problems, each of which augments the baseline OnlineKnapsack with additional constraints/controls. The formal models for these settings are presented in Section 2.

Instead of solving each problem in an ad hoc manner, however, our policies are all derived from a single underlying framework. In particular, our results can be summarized as follows:

Meta-Theorem. *Given an online allocation problem, we identify an appropriate offline benchmark, and give a simple*

online policy—based on solving a tractable optimization problem in each period—that gets constant regret compared with the benchmark (and thus, compared with the optimal policy).

In more detail, our approach is based on adaptively constructing a benchmark that has additional (but not necessarily full) information about future randomness. Next, in the spirit of online primal-dual methods, we use our benchmark to construct a feasible online policy. The centerpiece of our approach are the *Bellman inequalities*, which characterize what benchmarks are feasible and decompose the regret of an online policy into two distinct terms. The first, which we call the *Bellman loss*, arises from computational considerations, specifically, from requiring that the benchmark is tractable (instead of a dynamic program that may be intractable). The second, which we call the *information loss*, accounts for unpredictability across sample paths. Our policies trade off these two losses to get strong performance guarantees.

Our framework allows flexibility in choosing benchmarks. To understand why this is important, consider two common benchmarks for dynamic pricing: a controller has inventory B, and posts prices for T sequential customers, each of which has a random valuation. One common benchmark, known as the *offline* or *prophet* benchmark, considers a controller with full information of all randomness. It is easy to show that no online policy can get better than $\Omega(T)$ regret against this benchmark. An alternate benchmark, known as the *ex ante* or *fluid* benchmark, corresponds to replacing all random quantities with their expectations. Here again, no online policy can get better than $\Omega(\sqrt{T})$ regret (Vera and Banerjee 2020). Our approach, however, lets us identify benchmarks that have $O(1)$ regret for all our settings.

Prophet and fluid benchmarks are also widely used in adversarial models of online allocation, leading to algorithms with worst-case guarantees. In contrast, we consider stochastic inputs, and consequently get much stronger guarantees. In particular, all our guarantees are parametric and depend explicitly on the distributions and problem primitives (i.e., constant parameters defining the instance). All our policies, however, have regret that is independent of the horizon and budgets.

2. Preliminaries and Overview

2.1. Problem Settings and Results

We illustrate our framework by developing low-regret algorithms for the following problems:

Online stochastic knapsack. This serves as a baseline for our other problems. The controller has an initial resource budget B, and items arrive sequentially over T periods. Each item has a random type j, which corresponds to a known resource requirement (or weight) w_j and a random reward R_j. In period $t = T, T-1, \ldots, 1$ (where t denotes the time-to-go), we assume the arriving type is drawn from a finite set $[n]$ from some known distribution $\mathbf{p} = (p_1, \ldots, p_n)$. At the start of each period, the controller observes the type of the arriving item, and must decide to accept or reject the item. The expected reward from selecting a type-j item is $r_j = \mathbb{E}[R_j]$.

Online probing. As before, an arriving type j has known expected reward r_j, but unknown realized reward R_j. Now the controller has the additional option of probing each request to observe the realization, and then accept/reject the item based on the revealed reward. The controller can also choose to accept the item without probing. In addition to the resource budget B, the controller has an additional probing budget B_p that limits the number of arrivals that can be probed. This introduces a trade-off between depleting the resource budget B and probing budget B_p. We assume here that R_j has finite support $\{r_{jk}\}_{k\in[m]}$ of size m, and define $q_{jk} := \mathbb{P}[R_j = r_{jk}]$ for $k \in [m]$. Note this reduces to OnlineKnapsack when either $B_p \geq T$ or $B_p = 0$.

Dynamic pricing. The controller has an initial inventory $B \in \mathbb{N}^d$ for d different resources. There are n types of customers, where a customer of type j requests a specific subset $A_j \in \{0,1\}^d$ of resources, and has private valuation $R^t \sim F_j$. In each period t, the controller observes the customer type $j \in [n]$, and if sufficient resources are available, posts a price (fare) f from a finite set $\{f_{j1}, \ldots, f_{jm}\}$. The customer then purchases iff $R^t > f$. The vectors A_j and valuation functions $(F_j : j \in [n])$ are known, but otherwise arbitrary. More generally, our technique handles probabilistic customer-choice models, where a customer, when presented with a price menu over bundles, picks a random bundle via some known distribution (which may depend on the menu).

Knapsack with distribution learning. We return to the OnlineKnapsack setting where items of type $j \in [n]$ have weight w_j and random reward R_j. Now, however, the controller is unaware of the distribution of R_j, and must learn it from observations. In period t, the controller observes the arrival type j, and decides to accept/reject based on observed rewards up to time t. We consider two feedback models: *full feedback*, where the controller observes R_j regardless of whether the item is accepted or rejected, and *censored feedback*, where the controller only observes rewards of accepted items. For the latter (which is sometimes referred to as online contextual bandits with knapsacks), we assume the rewards R_j have sub-Gaussian tails (Boucheron et al. 2013, section 2.3).

Benchmarks and guarantees. Our framework, RABBI (for resolve and act based on the Bellman inequalities; see Section 3.2) is based on comparing two

controllers: Offline, which acts optimally given future information; and a nonanticipative controller Online, which tries to follow Offline. Both start in the same initial state S^T. We denote v^{off} as the expected total reward collected by Offline acting optimally (i.e., according to a Bellman equation) given its information structure. In contrast, Online uses a nonanticipative policy π that maps current states to actions, resulting in a total expected reward v_π^{on}.

Let π_R denote the online policy produced by our rabbi framework, and π denote any nonanticipative policy. Then the expected regret of π_R relative to the chosen offline benchmark is

$$\mathbb{E}[\text{Regret}] := v^{\text{off}} - v_{\pi_R}^{\text{on}} \geq \max_\pi [v_\pi^{\text{on}}] - v_{\pi_R}^{\text{on}}.$$

The last inequality, which follows from the fact that $v_\pi^{\text{on}} \leq v^{\text{off}}$ for any pair of benchmark and online policies, emphasizes that the regret is a bound on the additive gap with respect to (w.r.t.) the best online policy.

For all of these problems, we use the rabbi framework to identify an appropriate benchmark with respect to which we get the following guarantees. First, for the OnlineKnapsack, we recover a result proved in Arlotto and Gurvich (2019) and Vera and Banerjee (2020).

Theorem 1 (Theorem 1 in Arlotto and Gurvich 2019)**.** *For known reward distributions with finite mean, an online policy based on the* rabbi *framework obtains regret that depends only on the primitives* $(n, \mathbf{p}, \mathbf{r}, \mathbf{w})$*, but is independent of the horizon length* T *and resource budget* B*.*

Theorem 1 builds intuition for using rabbi in more complex settings. In particular, the benchmark used in Theorem 1 is the full-information prophet, which is too loose for obtaining constant regret in the remaining settings (pricing, probing, and bandits; see Example 1). This is where our framework helps in guiding the choice of the right benchmark. In particular, we obtain the following results.

Theorem 2 (Online Probing)**.** *For reward distributions with finite support of size* m*, an online-probing policy based on the* rabbi *framework (Algorithm 2) obtains regret that depends only on* $(n, m, \mathbf{q}, \mathbf{p}, \mathbf{r})$*, but is independent of horizon length* T*, resource budget* B *and probing budget* B_p*.*

Theorem 3 (Dynamic Pricing)**.** *For any reward distributions* $(F_j : j \in [n])$ *and prices* $\mathbf{f}$*, a pricing policy based on the* rabbi *framework (Algorithm 3) obtains regret that depends only on* $(A, \mathbf{f}, F_1, \ldots, F_n)$*, but is independent of horizon length* T *and initial budget levels* $B \in \mathbb{N}^d$*.*

The result for dynamic pricing also extends naturally to resource bundles and general customer-choice models (see Section 5.5 and Theorem 6 therein).

For the bandit settings, we define a separation parameter $\delta = \min_{j \neq j'} \mathbb{E}[R_j]/w_j - \mathbb{E}[R_{j'}]/w_{j'}$; this is only for our bounds, and is not known to the algorithm.

Theorem 4 (Knapsack with Distribution Learning)**.** *Assuming the reward distributions are sub-Gaussian, in the full feedback setting, a policy based on the* rabbi *framework (Algorithm 5) obtains regret that depends only on the primitives* $(n, \mathbf{p}, \mathbf{r}, \mathbf{w}, \delta)$ *and is independent of the horizon length* T *and knapsack capacity* B*.*

The last result can also be used as a black box for the censored feedback setting to get an $O(\log T)$ regret guarantee (see Corollary 1 in Section 6.3).

2.2. Overview of Our Framework

We develop our framework in the full generality of MDPs in Section 3. To give an overview and gain insight into the general version, we use OnlineKnapsack as a warm-up. A schema for the framework is provided in Figure 1.

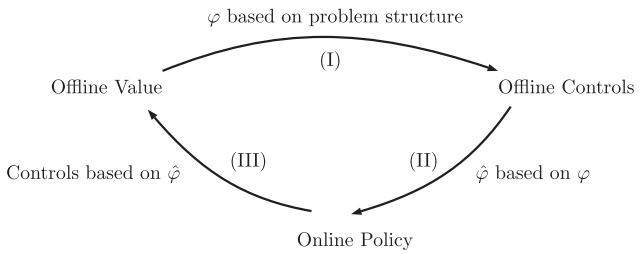
In the OnlineKnapsack problem, at any time-to-go t, let $Z_j^t \in \mathbb{N}$ denote the (random) number of type-j arrivals in the remaining t periods. Recall rewards of type j arrivals have expected value $r_j := \mathbb{E}[R_j]$. Define Offline to be a controller that knows Z^t for all t in advance. The total reward collected by Offline can be written as an integer linear program:

$$\begin{aligned} V(t, b | Z^t) &= \max_{\mathbf{x}_a \in \mathbb{N}^n} \{\mathbf{r}'\mathbf{x} : \mathbf{w}'\mathbf{x}_a \leq b, \mathbf{x}_a \leq Z^t\} \\ &= \max_{\mathbf{x}_a, \mathbf{x}_r \in \mathbb{N}^n} \{\mathbf{r}'\mathbf{x}_a : \mathbf{w}'\mathbf{x}_a \leq b, \mathbf{x}_a + \mathbf{x}_r = Z^t\}. \quad (1) \end{aligned}$$

The function $V(\cdot|Z^t)$ is thus Offline's value function (Figure 1), where the notation $|Z^t$ emphasizes that V is conditioned on Z^t. Moreover, for every j, the variables $x_{a,j}, x_{r,j}$ represent action summaries: the number of type-j arrivals accepted and rejected, respectively.

The value $V(\cdot|Z^t)$ can also be represented via Bellman equations. Specifically, at time-to-go t, assuming

Figure 1. The rabbi Framework



Notes. We first define Offline's value function by specifying access to future information. Next, we identify a tractable relaxation φ for Offline's value under this same information structure (step I). Finally, we introduce a nonanticipative estimate $\hat{\varphi}$ for φ, and use it to design online controls (step II). The resulting online policy is evaluated against Offline's value (step III).

OFFLINE has budget b and the arriving type is ξ, the value function obeys the Bellman equation:

$$V(t,b|Z^t) = \max\Big\{\big[r_{\xi^t} + V(t-1, b - w_{\xi^t}|Z^{t-1})\big]\mathbb{1}w_{\xi^t} \le b, \\ V(t-1,b|Z^{t-1})\Big\}, \quad \forall t, b, \xi^t.$$

Next, consider the linear programming relaxation for $V(t,b)$:

$$\varphi(t,b|Z^t) := \max_{\mathbf{x}_a,\mathbf{x}_r \ge 0} \{\mathbf{r}'\mathbf{x}_a : \mathbf{w}'\mathbf{x}_a \le b, \mathbf{x}_a + \mathbf{x}_r = Z^t\}.$$

It is clear that φ is more tractable than V, and that it approximates V up to an integrality gap. However, φ does not obey a Bellman equation. To circumvent this, we introduce the notion of Bellman inequalities, wherein we require that φ satisfies Bellman-like conditions for most sample paths. Formally, for some random variables L_B, we want φ to satisfy

$$\varphi(t,b|Z^t) \le \max\Big\{\big[r_{\xi^t} + \varphi(t-1, b - w_{\xi^t}|Z^{t-1})\big]\mathbb{1}_{\{w_{\xi^t} \le b\}}, \\ \varphi(t-1,b|Z^{t-1})\Big\} + L_B(t,b).$$

Note that, if $\mathbb{E}[L_B(t,b)]$ is small, with expectation taken over Z^t, then φ almost satisfies the Bellman equations. We henceforth refer to φ as a relaxed value for V and L_B as the Bellman loss.

Establishing that actions derived from φ are nearly optimal for OFFLINE accomplishes step (I) in Figure 1. For step (II), we want to emulate OFFLINE by estimating φ based on current information. A natural estimate is obtained by taking expectations over future randomness, to get the following:

$$\hat{\varphi}(t,b) := \max_{\mathbf{y}_a,\mathbf{y}_r \ge 0} \{\mathbf{r}'\mathbf{y}_a : \mathbf{w}'\mathbf{y}_a \le b, \mathbf{y}_a + \mathbf{y}_r = \mathbb{E}[Z^t]\}.$$

Note that $\hat{\varphi}$ does not approximate V or φ up to a constant additive error (Vera and Banerjee 2020); however, $\hat{\varphi}$ can be used as a predictor for the action taken by OFFLINE. Specifically, at time t with current budget b, RABBI first computes $\hat{\varphi}(t,b)$ and then interprets the solution $\mathbf{y}$ as a score for each action (here, accept/reject). We show that taking the action with the highest score (i.e., action $\arg\max_{u\in\{a,r\}}\{y_{\xi^t,u}\}$) guarantees that ONLINE and OFFLINE play the same action with high probability. Whenever OFFLINE and ONLINE play different actions, we incur a loss, which we refer to as the information loss, as it quantifies how having less information impacts ONLINE's actions. This process of using $\hat{\varphi}$ to derive actions is represented as step (III) in Figure 1.

2.2.1. Toward a General Framework. For all the problems in Section 2.1, our approach uses a similar three-step process, wherein we choose an OFFLINE benchmark, identify relaxed value φ via appropriate optimization problem, and get an online policy based on estimate $\hat{\varphi}$. Consequently, we refer to our framework as RABBI, which stands for resolve and act based on Bellman inequalities.

Our work builds on constant-regret policies for multidimensional packing (Vera and Banerjee 2020), and more general online optimization problems (Banerjee and Freund 2020). The techniques developed in these works, however, have two fundamental shortcomings that prevent them from addressing the settings we consider:

- Use of full-information benchmarks: Existing works (Arlotto and Gurvich 2019, Banerjee and Freund 2020, Vera and Banerjee 2020) use the full-information benchmark, which is too loose for our settings. Indeed, for probing/pricing/learning settings, no algorithm can have constant regret compared with the full-information benchmark (see Example 1).
- Explicit value function characterizations: The optimization problem in Equation (1) has a closed-form solution, which was used explicitly by Arlotto and Gurvich (2019), Vera and Banerjee (2020), and Banerjee and Freund (2020). This does not extend to more complex settings.

Our framework in this work resolves these shortcomings in a structured way, allowing us to get provably near-optimal algorithms for several canonical resource-allocation problems. Moreover, we do so via a generalized notion of information-augmented benchmarks, and our decomposition of the regret into the information loss (capturing randomness in inputs) and Bellman loss (capturing limited computational power). This flexibility helps greatly in the design of our algorithms.

2.3. Related Work

Our approach has commonalities with two closely related approaches:

Prophet inequalities and ex ante relaxations: A well-studied framework for obtaining performance guarantees for heuristics policies is to compare against a full-information agent, or prophet. This line of work focuses on competitive-ratio bounds (see Kleinberg and Weinberg 2012, Correa et al. 2017, Düetting et al. 2017 for overviews of the area). In particular, Correa et al. (2017) obtains a multiplicative guarantee for dynamic posted pricing with a single item under worst-case distribution. A related line of work considers the use of ex ante linear program (LP) relaxations (Alaei 2014, Buchbinder et al. 2014) for obtaining worst-case competitive guarantees in online packing problems. In contrast, we obtain an additive guarantee for multiple items in a parametric setting.

MDP dual relaxations: A standard way to get bounds on MDPs is via information-relaxations, which at a

high level, create benchmarks by endowing OFFLINE with additional information, while forcing it to pay a penalty for using this information. Brown et al. (2010) and Balseiro and Brown (2019) use this in a dual-fitting approach to construct performance bounds for greedy algorithms in different problems. In contrast, our framework is similar to a primal-dual approach: we adaptively construct our relaxations, and derive controls directly from them. We compare the two approaches in more detail in Online Appendix EC.5. Moreover, the different problems we apply RABBI to each have a large body of prior work.

2.3.1. Online Packing. There is a long line of work on the baseline OnlineKnapsack and generalizations. A notable work in this line is Jasin and Kumar (2012), who gives a policy with constant expected regret when the problem instance is far from a set of certain nondegenerate instances. This inefficiency, though, is fundamental, since they use the ex ante (or fluid) benchmark, which has $\Omega(\sqrt{T})$ under nondegeneracy. More recently, Bumpensanti and Wang (2020) partially extend the result of Arlotto and Gurvich (2019) for more general packing problems; however, their policy only gives constant regret under independent and identically distributed (i.i.d.) Poisson arrivals, and requires the system to be scaled linearly (i.e., B grows proportional to T). In contrast, Arlotto and Gurvich (2019) (one dimension) and Vera and Banerjee (2020) (multiple dimensions) provide constant-regret policies with no assumption on the scaling. The approach in the latter is further generalized in Banerjee and Freund (2020) to handle more complex problems including bin packing and quality of service (QOS) constraints. See Vera and Banerjee (2020), Banerjee and Freund (2020) for more discussion and references.

2.3.2. Probing. Approximation algorithms have been developed for offline probing problems, both under budget constraints (Gupta and Nagarajan 2013) and probing costs (Weitzman 1979, Singla 2018). Another line of work pursues tractable nonadaptive constant-factor competitive algorithms for this problem (Gupta et al. 2016). In terms of online adaptive algorithms, Chugg and Maehara (2019) introduce an algorithm with bounded competitive ratio in an adversarial setting.

2.3.3. Dynamic Posted Pricing. This is a canonical problem in operations management with a vast literature (see Talluri and Van Ryzin 2006 for an overview). Much of this literature focuses on asymptotically optimal policies in regimes where the inventory B and/or horizon T grow large. When B and T are scaled together by a factor k, there are known algorithms with regret that scales as $O(\sqrt{k})$ or $O(\log(k))$, depending on assumptions on the primitives (e.g., smoothness of the demand with price) (Jasin 2014). There is also vast literature on pricing when the demand function is not known and has to be learned (Chen et al. 2019). Finally, under adversarial arrivals, Babaioff et al. (2015) provide a policy with $O((B\log T)^{2/3})$ regret under adversarial inputs, as opposed to our $O(1)$ guarantee under stochastic inputs.

2.3.4. Knapsack with Learning. Multiarmed bandit problems have been widely studied, and we refer to Bubeck et al. (2012, 2013) for an overview. Bandit problems with combinatorial constraints on the arms are known as *bandits with knapsacks* (Badanidiyuru et al. 2018), and the generalization where arms arrive online is known as *contextual bandits with knapsacks* (Badanidiyuru et al. 2014, Agrawal and Devanur 2016). Results in this literature typically study worst-case distributions. We, in contrast, pursue parametric regret bounds that explicitly depend on the (unknown) discrete distribution. Closest to our work is Wu et al. (2015), who provide a upper confidence bounds (UCB)-based algorithm that gets $O(\sqrt{T})$ regret (in contrast, we get $O(\log T)$ regret for the same setting).

3. Approximate Control Policies via the Bellman Inequalities

In this section, we describe our general framework. Before proceeding, we introduce some notation. We work an underlying probability space $(\Omega, \Sigma, \mathbb{P})$, and for any event $\mathcal{B} \subseteq \Omega$, we denote its complement by $\mathcal{B}^c$. We use boldface letters to indicate vector-valued variables (e.g. $\mathbf{p}, \mathbf{w}$, etc.), and capital letters to denote matrices and/or random variables. For an optimization problem (P), we use P to denote its optimal value. When using LP formulations with decision variables $\mathbf{x}$, we interchangeably use $x_{ij} = x(i,j)$ to denote the (i, j)th component of $\mathbf{x}$.

3.1. Offline Benchmarks and Bellman Inequalities

We consider an online decision-making problem with state space $\mathcal{S}$ and action space $\mathcal{U}$, evolving over periods $t = T, T-1, \ldots, 1$; here T denotes the horizon, and t is the time to-go. In any period t, the controller first observes a random arrival $\xi^t \in \Xi$, following which it must choose an action $u \in \mathcal{U}$. For system-state $s \in \mathcal{S}$ at the beginning of period t, and random arrival $\xi \in \Xi$, an action $u \in \mathcal{U}$ results in a reward $\mathcal{R}(s, \xi, u)$, and transition to the next state $\mathcal{T}(s, \xi, u)$. We assume both reward and future state are random variables whose realizations are determined for every u given ξ. This assumption is for ease of exposition only; our results can be extended to hold when rewards or transitions are random given ξ.

The feasible actions for state s and input ξ correspond to the set $\{u \in \mathcal{U} : \mathcal{R}(s, \xi, u) > -\infty\}$. We assume that this feasible set is nonempty for all $s \in \mathcal{S}, \xi \in \Xi$,

and that the maximum reward is bounded, that is, $\sup_{s\in\mathcal{S},\xi\in\Xi,u\in\mathcal{U}} \mathcal{R}(s,\xi,u) < \infty$.

The MDP described previously induces a natural filtration $\mathcal{F}$, with $\mathcal{F}_t = \sigma(\{\xi^\tau : \tau \geq t\})$; a nonanticipative policy is one that is adapted to $\mathcal{F}_t$. We allow Offline to use a richer information filtration $\mathcal{G}$, where $\mathcal{G}_t \supseteq \mathcal{F}_t$. Note that since t denotes the time-to-go, we have $\mathcal{G}_{t-1} \supseteq \mathcal{G}_t$. Henceforth, to keep track of the information structure, we use the notation $f(\cdot|\mathcal{G}_t)$ to clarify that a function f is measurable with respect to the sigma field $\mathcal{G}_t$.

Given any filtration $\mathcal{G}$, Offline is assumed to play the optimal policy adapted to $\mathcal{G}$, hence Offline's value function is given by the following Bellman equation:

$$V(t,s|\mathcal{G}_t) = \max_{u\in\mathcal{U}}\{\mathcal{R}(s,\xi^t,u) + \mathbb{E}\big[V\big(t-1,\mathcal{T}\big(s,\xi^t,u\big)|\mathcal{G}_{t-1}\big)|\mathcal{G}_t\big]\}, \quad (2)$$

with the boundary condition $V(0,\cdot) = 0$. We denote the expected value as $v^{\text{off}} := \mathbb{E}[V(T,S^T|\mathcal{G}_T)]$. Note that v^{off} is an upper bound on the performance of the optimal nonanticipative policy.

We present a specific class of filtration (generated by augmenting the canonical filtration) that suffice for our applications (see Figure 2 for an illustration of the definition).

Definition 1 (Canonical Augmented Filtration)**.** Let $G_\Theta := (G_\theta : \theta \in \Theta)$ be a set of random variables. The canonical filtration w.r.t. G_Θ is

$$\mathcal{G}_t = \sigma\Big(\{\xi^l : l \geq t\} \cup G_\Theta\Big) \supseteq \mathcal{F}_t.$$

The richest augmented filtration is the full-information filtration, wherein $\mathcal{G}_t = \mathcal{F}_1$ for all t, that is, the canonical filtration with $G_\Theta = (\xi^t : t \in [T])$. As $\mathcal{G}_t$ gets coarser, the difference in performance between Offline and Online decreases. Indeed, when $\mathcal{G} = \mathcal{F}$, then Equation (2) reduces to the Bellman equation for the value function of an optimal nonanticipative policy:

$$V(t,s|\mathcal{F}_t) = \max_{u\in\mathcal{U}}\{\mathcal{R}(s,\xi^t,u) + \mathbb{E}\big[V\big(t-1,\mathcal{T}\big(s,\xi^t,u\big)|\mathcal{F}_{t-1}\big)\big]\}, \quad V(0,\cdot,\cdot) = 0,$$

where the expectation is taken with respect to the next period's input ξ^{t-1}.

Example 1 (Full Information Is Too Loose)**.** Consider a dynamic pricing instance with $n = d = 1$, prices $\mathbf{f} = (1,2)$, and valuation distribution $\mathbb{P}[R^t = 1+\varepsilon] = p$ and $\mathbb{P}[R^t = 2+\varepsilon] = 1-p$. When $B = T$, the optimal policy always posts a price that maximizes $(f \cdot \mathbb{P}[R^t > f])$. If $p \geq 1/2$, then the optimal policy (DP) always posts price $f = 1$ and has expected reward T. On the other hand, full information can post price $R^t - \varepsilon$ at time t and extract full surplus $v^{\text{off}} = \sum_t \mathbb{E}[R^t - \varepsilon] = T(2-p)$. Thus, the regret against full information must grow as $\Omega(T)$. This example is not pathological; the same behavior persists even in random instances (see Section 5.4).

We are now ready to introduce the notion of relaxed value φ and Bellman inequalities. Intuitively, φ is almost defined by a dynamic-programming recursion; quantitatively, whenever φ does not satisfy the Bellman equation, we incur an additional loss L_B, which we denote the Bellman loss.

Definition 2 (Bellman Inequalities)**.** The family of random variables $\{\varphi(t,s)\}_{t,s}$ satisfies the Bellman inequalities w.r.t. filtration $\mathcal{G}$ and random variables $\{L_B(t,s)\}_{t,s}$ if $\varphi(t,\cdot)$ and $L_B(t,\cdot)$ are $\mathcal{G}_t$-measurable for all t and the following conditions hold:

1. Initial ordering: $\mathbb{E}[V(T,S^T)|\mathcal{G}_T] \leq \varphi(T,S^T|\mathcal{G}_T)$.
2. Monotonicity: $\forall s \in \mathcal{S}, t \in [T]$,

$$\varphi(t,s|\mathcal{G}_t) \leq \max_{u\in\mathcal{U}}\{\mathcal{R}(s,\xi^t,u) + \mathbb{E}\big[\varphi\big(t-1,\mathcal{T}\big(s,\xi^t,u\big)|\mathcal{G}_{t-1}\big)|\mathcal{G}_t\big]\} + L_B(t,s). \quad (3)$$

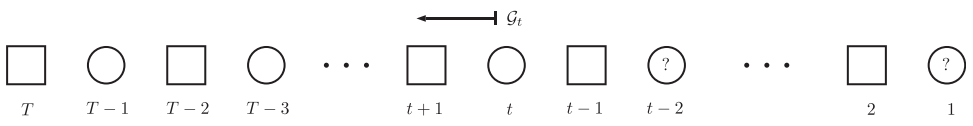
3. Terminal condition: $\varphi(0,s) = 0\ \forall\, s \in \mathcal{S}$.

We refer to φ and L_B as the relaxed value and Bellman loss pair with respect to $\mathcal{G}$, and use $|\mathcal{G}_t$ to remind the reader that we need the information contained in $\mathcal{G}_t$ to evaluate $\varphi(t,s)$.

Given any φ, monotonicity holds trivially with $L_B = \varphi$ (but leads to poor performance guarantees). On the other hand, φ (which may be intractable) is the only value function guaranteeing $L_B = 0$. The crux of our approach is to identify a good φ balances the loss and tractability.

A special case is when the Bellman loss is 0 over sample paths in some chosen set.

Figure 2. Illustration of Definition 1



Notes. In online probing (see Section 4), arrivals first reveal their public type, then the controller chooses an action (accept/probe/reject), and then the private type (true reward) is revealed. Squares (respectively, circles) represent public (respectively, private) information. The filtration $\mathcal{G}$ used by RABBI comprises of all public types, that is, $G_\Theta = (\xi^\theta : \xi^\theta$ is a public type). At time t, Offline knows all the information thus far (to the left and including t), plus the future squares.

Definition 3 (Exclusion Sets)**.** A set $\mathcal{B}(t,s)$ is an exclusion set if we can write the Bellman loss as $L_B(t,s) = r_\varphi \mathbb{1}_{\mathcal{B}(t,s)}$ for some constant $r_\varphi > 0$ and events $\mathcal{B}(t,s) \subseteq \Omega$.

If the Bellman loss can be defined with exclusion sets, then from Definition 2 (monotonicity), we obtain the condition $\varphi(t,s|\mathcal{G}_t) \le \max_{u\in\mathcal{U}}\{\mathcal{R}(s,\xi^t,u) + \mathbb{E}[\varphi(t-1,\mathcal{T}(s,\xi^t,u)|\mathcal{G}_{t-1})|\mathcal{G}_t]$, that is, monotonicity is satisfied for all realizations $\omega \in \Omega$ except for those in the exclusion set $\mathcal{B}(t,s)$.

To build intuition, we specify the Bellman inequalities for our baseline OnlineKnapsack. For this end, we first need the following lemma characterizing the sensitivity of LP solutions.

Lemma 1. *Consider an LP* $(P[\mathbf{d}]) : \max\{\mathbf{r}'\mathbf{x} : M\mathbf{x} = \mathbf{d}, \mathbf{x} \ge 0\}$, *where* $M \in \mathbb{R}^{m\times n}$ *is an arbitrary constraint matrix. If* $\bar{\mathbf{x}}$ *solves* $(P[\mathbf{d}])$ *and* $\bar{x}_j \ge 1$ *for some* j, *then* $P[\mathbf{d}] = r_j + P[\mathbf{d} - M_j]$.

Proof. By assumption, the optimal value of $(P[\mathbf{d}])$ remains unchanged if we add the inequality $x_j \ge 1$. Therefore, we have $P[\mathbf{d}] = \max\{\mathbf{r}'(\mathbf{x}+\mathbf{e}_j) : M(\mathbf{x}+\mathbf{e}_j) = \mathbf{d}, \mathbf{x} \ge 0\}$. □

Lemma 1 lets us divide $P[\mathbf{d}]$ into two summands: the immediate reward r_j and the future reward $P[\mathbf{d} - M_j]$. This has the flavor of dynamic programming we need for defining the Bellman loss.

Example 2 (Bellman Loss for Baseline Setting)**.** For the baseline OnlineKnapsack, discussed in Section 2.2, we chose the full-information filtration $\mathcal{G}_t = \mathcal{F}_1$ for all t so that $\varphi(t,b|\mathcal{G}_t) := \max_{\mathbf{x}\ge 0}\{\mathbf{r}'\mathbf{x}_a : \mathbf{w}'\mathbf{x}_a \le b, \mathbf{x}_a + \mathbf{x}_r = Z^t\}$. We define the exclusion sets as

$$\mathcal{B}(t,b) = \{\omega \in \Omega : \nexists \mathbf{x} \text{ solving } \varphi(t,b) \text{ s.t. } x(\mathtt{a},\xi^t) \ge 1 \text{ or } x(\mathtt{r},\xi^t) \ge 1\}.$$

By Lemma 1, outside the exclusion sets $\mathcal{B}(t,b)$, monotonicity holds with zero Bellman Loss, that is,

$$\varphi(t,s|\mathcal{G}_t) \le \max_{u\in\mathcal{U}}\{\mathcal{R}(s,\xi^t,u) + \mathbb{E}[\varphi(t-1,\mathcal{T}(s,\xi^t,u)|\mathcal{G}_{t-1})|\mathcal{G}_t]\} \quad \forall \omega \notin \mathcal{B}(t,s).$$

Moreover, for our choice of φ, since the optimal solution sorts items by r_j/w_j, we have that the maximum loss outside the exclusion set is bounded by $r_\varphi \le \max_{j,i}\{w_i r_j/w_j - r_i\}$, which depends only on the primitives. Thus, Definition 2 is satisfied with Bellman loss $L_B(t,b) = r_\varphi \mathbb{1}_{\mathcal{B}(t,b)}$.

To generalize this, we need two definitions. First, we define the maximum Bellman loss as follows.

Definition 4 (Maximum Loss)**.** For a given relaxation φ, the maximum loss is given by

$$r_\varphi := \max_{t,s,u:\mathcal{R}(s,\xi^t,u)>-\infty}\{\varphi(t,s|\mathcal{G}_t) - (\mathcal{R}(s,\xi^t,u) + \mathbb{E}[\varphi(t-1,\mathcal{T}(s,\xi^t,u)|\mathcal{G}_{t-1})|\mathcal{G}_t])\}.$$

Next, note that the optimal action in the right-hand side (RHS) of Equation (3) need not be unique, and indeed the inequality can be satisfied by multiple actions. For given φ and L_B, we define the following.

Definition 5 (Satisfying Actions)**.** Given a filtration $\mathcal{G}$ and relaxed value φ, we say that u is a satisfying action for state s at time t if

$$\varphi(t,s|\mathcal{G}_t) \le \mathcal{R}(s,\xi^t,u) + \mathbb{E}[\varphi(t-1,\mathcal{T}(s,\xi^t,u)|\mathcal{G}_{t-1})|\mathcal{G}_t] + L_B(t,s). \tag{4}$$

At any time t and state $s \in \mathcal{S}$, any action in $\arg\max_{u\in\mathcal{U}}\{\mathcal{R}(s,\xi^t,u) + \mathbb{E}[\varphi(t-1,\mathcal{T}(s,\xi^t,u)|\mathcal{G}_{t-1})|\mathcal{G}_t]\}$ is always a satisfying action (see monotonicity in Definition 2). Moreover, to identify a satisfying action, we must know $\mathcal{G}_t$. We now have the following proposition.

Proposition 1. *Consider a relaxation* φ *and Bellman loss* L_B *that satisfy the Bellman inequalities w.r.t. filtration* $\mathcal{G}$. *Let* $(S^t, t \in [T])$ *denote the state trajectory under a policy that, at time* t, *takes any satisfying action* $U^t = U^t(S^t|\mathcal{G}_t)$. *Then,*

$$\mathbb{E}[V(T,S^T|\mathcal{G}_T)] - \mathbb{E}\left[\sum_{t=1}^T \mathcal{R}(S^t,\xi^t,U^t)\right] \le \mathbb{E}\left[\sum_{t=1}^T L_B(t,S^t|\mathcal{G}_t)\right].$$

Proof. From the monotonicity condition in the Bellman inequalities (Definition 2), and the definition of a satisfying action (Definition 5), we have, for all time t, that

$$\varphi(t,S^t|\mathcal{G}_t) \le \mathbb{E}[\mathcal{R}(S^t,\xi^t,U^t) + \varphi(t-1,S^{t-1}|\mathcal{G}_{t-1}) + L_B(t,S^t|\mathcal{G}_t)|\mathcal{G}_t].$$

Iterating this inequality over t, we get $\varphi(T,S^T|\mathcal{G}_T) \le \sum_{t=1}^T \mathbb{E}[\mathcal{R}(S^t,\xi^t,U^t) + L_B(t,S^t|\mathcal{G}_t)|\mathcal{G}_t]$. Finally, by the initial ordering condition, we have $\mathbb{E}[V(T,S^T)|\mathcal{G}_T] \le \varphi(T,S^T|\mathcal{G}_T)$. □

Proposition 1 shows that a policy that always plays a satisfying action U^t approximates the performance of Offline up to an additive gap given by the total Bellman loss $\mathbb{E}[\sum_{t=1}^T L_B(t,S^t|\mathcal{G}_t)]$. More importantly, it suggests that Online should try to track Offline by guessing and playing a satisfying action U^t in each period. We next illustrate how Online can generate such guesses.

3.2. From Relaxations to Online Policies

Suppose we are given an augmented canonical filtration $\mathcal{G}_t = \sigma(\{\xi^l : l \ge t\} \cup G_\Theta)$, and assume that the relaxed value φ can be represented as a function of the random variables $\{\xi^l : l \ge t\} \cup G_\Theta$ as $\varphi(t,s|\mathcal{G}_t) = \varphi(t,s;f_t(\xi^T,\dots,\xi^t,G_\Theta))$. In particular, we henceforth focus on a special case where φ is expressed as the solution of an optimization problem:

$$\varphi(t,s;f_t(\xi^T,\dots,\xi^t,G_\Theta)) = \max_{\mathbf{x}\in\mathbb{R}^{\mathcal{U}\times\Xi}}\{h_t(\mathbf{x};s,f_t(\xi^T,\dots,\xi^t,G_\Theta)) : g_t(\mathbf{x};s,f_t(\xi^T,\dots,\xi^t,G_\Theta)) \le 0\} \tag{5}$$

The decision variables give action summaries: for given state s and time t, $x_{u,\xi}$ represents the number of times action u is taken for input ξ in remaining periods. We can also interpret $x_{u,\xi}$ as a score for action u when input ξ is presented. Now, to get a nonanticipative policy, a natural projection of $\varphi(t,s|\mathcal{G}_t)$ on the filtration $\mathcal{F}$ is given via the following optimization problem:

$$\begin{aligned}\hat{\varphi}(t,s|\mathcal{F}_t) &= \varphi\big(t,s;\mathbb{E}\big[f_t\big(\xi^T,\dots,\xi^t,G_\Theta\big)|\mathcal{F}_t\big]\big)\\ &= \max_{\mathbf{y}\in\mathbb{R}^{\mathcal{U}\times\Xi}}\big\{h_t\big(\mathbf{y};s,\mathbb{E}\big[f_t|\mathcal{F}_t\big]\big) : g_t\big(\mathbf{y};s,\mathbb{E}\big[f_t|\mathcal{F}_t\big]\big)\le 0\big\}.\end{aligned} \tag{6}$$

The solution of this optimization problem gives action summaries (or scores) $\mathbf{y}$. The main idea of the RABBI algorithm is to play the action with the highest score.

Algorithm 1 RABBI (Resolve and Act Based on Bellman Inequalities)

Input: Access to functions f_t such that $\varphi(t,s|\mathcal{G}_t) = \varphi(t,s;f_t(\xi^T,\dots,\xi^t,G_\Theta))$.
Output: Sequence of decisions $\hat{U}^t$ for ONLINE.
1: Set S^T as the given initial state
2: **for** $t = T,\dots,1$ **do**
3: Compute $\hat{\varphi}(t,S^t) = \varphi(t,S^t;\mathbb{E}[f_t(\xi^T,\dots,\xi^t,G_\Theta)|\mathcal{F}_t])$ with associated scores $\mathbf{y} = \{y_{u,\xi}\}_{u\in\mathcal{U},\xi\in\Xi}$.
4: Given input ξ^t, choose the action $\hat{U}^t$ with the highest score y_{u,ξ^t}.
5: Collect reward $\mathcal{R}(S^t,\xi^t,\hat{U}^t)$; update state $S^{t-1}\leftarrow\mathcal{T}(S^t,\xi^t,\hat{U}^t)$.

Theorem 5. *Let* OFFLINE *be defined by an augmented filtration $\mathcal{G}_t$ as in Definition 1. Assume the relaxation $\varphi(t,s)$ satisfies the Bellman inequalities with loss L_B, and for all $(t,a)\in[T]\times\mathcal{S}$, let $\mathcal{Q}(t,s)\subseteq\Omega$ denote the set of sample paths where the action $\hat{U}^t$ taken by* RABBI *is not a satisfying action. If $(S^t, t\in[T])$ denotes the state trajectory under* RABBI*, then*

$$\begin{aligned}\mathbb{E}[\text{Regret}] &\le \mathbb{E}\left[\sum_t\big(r_\varphi\mathbb{1}_{\mathcal{Q}(t,S^t)} + \mathbb{1}_{\mathcal{Q}(t,S^t)^c}L_B\big(t,S^t\big)\big)\right]\\ &\le \sum_t\big(r_\varphi\mathbb{P}\big[\mathcal{Q}\big(t,S^t\big)\big] + \mathbb{E}\big[L_B\big(t,S^t\big)\big]\big).\end{aligned}$$

Remark 1 (Bellman and Information Loss)**.** The bound in Theorem 5 has two distinct summands: The information loss $\sum_t\mathbb{P}[\mathcal{Q}(t,S^t)]$ measures how often RABBI takes a nonsatisfying action due to randomness in sample paths; and the Bellman loss $\sum_t\mathbb{E}[L_B(t,S^t)])$ quantifies violations of the Bellman equations made under the pseudo value function φ.

The proof of Theorem 5 is based on the compensated coupling approach introduced in Vera and Banerjee (2020). The idea is to imagine simulating controllers OFFLINE and ONLINE with identical random inputs $(\xi^t : t\in[T])$, with ONLINE acting before OFFLINE. Moreover, suppose at some time t, both controllers are in the same state s. Recall that, for any given state s at time t, an action u is satisfying if OFFLINE's value does not decrease when playing u (Definition 5). If ONLINE chooses to play a satisfying action, then we can make OFFLINE play the same action, and consequently both move to the same state. On the other hand, if ONLINE chooses an action that is not satisfying, then the two trajectories may separate. We can avoid this, however, by compensating OFFLINE so that it agrees to take the same action as ONLINE. In particular, it is always sufficient to compensate OFFLINE by the maximum loss r_φ to ensure its reward does not decrease by following ONLINE. As a consequence, the (compensated) OFFLINE and ONLINE take the same actions, and thus their trajectories are coupled.

As an example, for OnlineKnapsack with budget $B = 2$, weights $w_j = 1\,\forall\, j$, and horizon $T = 5$, consider a sample path $\omega\in\Omega$ with rewards $(\xi^5,\xi^4,\xi^3,\xi^2,\xi^1) = (5,7,2,7,2)$. The sample path comprises three different types, and the sequence of actions $(\mathtt{r},\mathtt{a},\mathtt{r},\mathtt{a},\mathtt{r})$ (selecting the value 7 items) is optimal for OFFLINE, with total reward of 14. Suppose ONLINE, in period $t = 5$, wants to accept the item with reward $\xi^5 = 5$; then, OFFLINE is willing to follow this action if given a compensation of 2 (in addition to collecting reward 5). OFFLINE and ONLINE then start the next period $t = 4$ in the same state with budget 1, hence remain coupled.

Proof of Theorem 5. Denoting OFFLINE's state as $\bar{S}^t$, we have via Proposition 1 that $\forall\, t$:

$$\begin{aligned}\varphi\big(t,\bar{S}^t|\mathcal{G}_t\big) \le \mathbb{E}\big[&\mathcal{R}\big(\bar{S}^t,\xi^t,U^t\big) + \varphi\big(t-1,\bar{S}^{t-1}|\mathcal{G}_{t-1}\big)\\ &+ L_B\big(t,\bar{S}^t\big)|\mathcal{G}_t\big].\end{aligned}$$

Let us assume as the induction hypothesis that $\bar{S}^t = S^t$. This holds for $t = T$ by definition. At any time t and state S^t, if $\hat{U}^t$ is not a satisfying action for OFFLINE, then we have from the definition of the maximum loss (Definition 4) that:

$$\begin{aligned}r_\varphi \ge \varphi\big(t,S^t|\mathcal{G}_t\big) &- \mathcal{R}\big(S^t,\xi^t,\hat{U}^t\big)\\ &+ \mathbb{E}\big[\varphi\big(t-1,S^{t-1}|\mathcal{G}_{t-1}\big)|\mathcal{G}_t\big]\big) \quad \textit{almost surely.}\end{aligned}$$

Now to make OFFLINE take action $\hat{U}^t$ so as to have the same subsequent state as ONLINE, it is sufficient to compensate OFFLINE with an additional reward of r_φ. Specifically, we have

$$\begin{aligned}\varphi\big(t,S^t|\mathcal{G}_t\big) \le \mathbb{E}\big[&\mathcal{R}\big(S^t,\xi^t,\hat{U}^t\big) + \varphi\big(t-1,S^{t-1}|\mathcal{G}_{t-1}\big)\\ &+ r_\varphi\mathbb{1}_{\mathcal{Q}(t,S^t)} + \mathbb{1}_{\mathcal{Q}(t,S^t)^c}L_B\big(t,S^t\big)|\mathcal{G}_t\big].\end{aligned}$$

Finally, as in Proposition 1, we can iterate over t to obtain

$$\begin{aligned}\mathbb{E}\big[\varphi\big(T,S^T|\mathcal{G}_T\big)\big] \le \mathbb{E}\Bigg[&\sum_t\mathcal{R}\big(S^t,\xi^t,\hat{U}^t\big).\\ &+\sum_t\big(r_\varphi\mathbb{1}_{\mathcal{Q}(t,S^t)} + \mathbb{1}_{\mathcal{Q}(t,S^t)^c}L_B\big(t,S^t\big)\big)\Bigg].\end{aligned}$$

The first sum on the right-hand side corresponds exactly to Online's total reward using the rabbi policy. By the initial ordering property, $\mathbb{E}[V(T,S^T)] \le \mathbb{E}[\varphi(T,S^T)]$, and we get the result. □

4. Online Probing

We now apply our framework to online probing. Here, each arrival type j has an independent random reward $R_j \in \{r_{jk} : k \in [m]\}$ drawn with probabilities $\{q_{jk}\}$; $\mathbf{r}$ and $\mathbf{q}$ are known. We assume without loss of generality (w.l.o.g.) that $r_{j1} < r_{j2} < \ldots < r_{jm}$ and $r_{jm} > 0$. For ease of exposition, we assume that all arrivals have unit weights; our analysis however extends to general weights w_j. The controller may accept (a), reject (r), or probe (p) the arrival. Accepting a type-j item without probing results in expected reward of $\bar{r}_j := \sum_{k\in[m]} r_{jk}q_{jk}$. Probing reveals the realized reward, after which it can be accepted or rejected. The controller has a resource budget $B_h \in \mathbb{N}$ and a probing budget $B_p \in \mathbb{N}$. When an arrival is accepted (respectively, probed), we reduce B_h (respectively, B_p) by one.

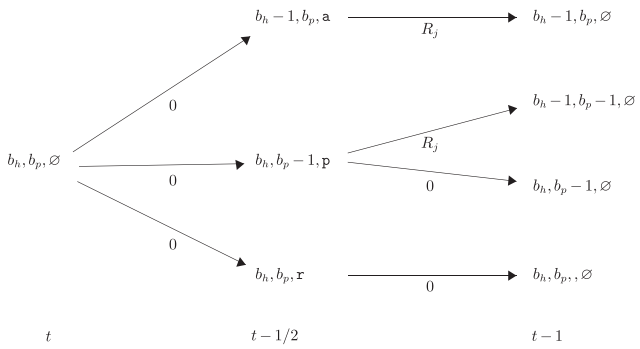
Formally, we view each time period $t \in \{T, T-1, \ldots, 1\}$ as comprising of a mini dynamic program with two stages $\{t, t-1/2\}$, driven by external random inputs $\xi^t \in [n]$ and $\xi^{t-1/2} \in [n]\times[m]$. In the first stage t, the controller observes the arriving request $\xi^t = j$, and chooses an action in $\{\mathtt{a},\mathtt{p},\mathtt{r}\}$; in the second stage $t-1/2$, the reward r_{jk} (or subtype $\xi^{t-1/2} = (j,k) \in [n]\times[m]$) is drawn with probability q_{jk}, and the available actions are $\{\mathtt{a},\mathtt{r}\}$ if the first-stage action is p, and $\emptyset$ otherwise. We augment the state space with a variable $\diamond$ that captures the first-stage decision (i.e., whether we accept/reject without probing or probe). The state space $\mathcal{S}$ of the controlled process is thus $\mathcal{S} = \{(b_h, b_p, \diamond) : b_h, b_p \in \mathbb{N}, \diamond \in \{\mathtt{a},\mathtt{p},\mathtt{r},\emptyset\}\}$, where b_h, b_p are the residual hiring and probing budgets. In first stage of each period, we set $\diamond = \emptyset$, and only collect rewards in the second stage in each period. See Figure 3 for an illustration.

4.1. Offline Benchmark and Online Policy for Probing

We now apply the rabbi framework for online probing.

4.1.1. Offline Benchmark. We define Offline to be the controller that knows the public types of all arrivals in advance (i.e., it knows Z_j^t, the number of type-j items that will arrive in the last t periods), but does not know the realization of the rewards (subtypes). Formally, Offline is endowed with the canonical filtration given by $\Theta = [T]$ and $G_\theta = \xi^\theta$ (see Definition 1): with t steps to go, Offline has the information filtration $\mathcal{G}_t = \sigma(\{\xi^t : t \in [T]\} \cup \{\xi^\tau : \tau \ge t\})$. Note that since Offline does not know the actual rewards, it still needs to solve a dynamic program to decide whether to probe an arrival.

Figure 3. Actions/Transitions in Online Probing in Periods t, $t-1/2$, and $t-1$, with Inputs $\xi^t = j$ and $\xi^{t-1/2} = R^j$



Notes. Numbers below the arrows represent the reward of a transition. At t, available actions are $\{\mathtt{a},\mathtt{p},\mathtt{r}\}$ (i.e., accept, probe, reject; from top to bottom). At $t-1/2$, if we chose to probe in the first stage (i.e., are in the middle state), then available actions are $\{\mathtt{a},\mathtt{r}\}$.

4.1.2. Relaxed Value Function. Since solving for Offline's optimal actions may be nontrivial, we next construct a relaxed value function φ, using the following LP parametrized by $(b_h, b_p, \mathbf{z}) \in \mathbb{N}^2 \times \mathbb{R}^n_{\ge 0}$:

$$
\begin{aligned}
&\big(P[b_h, b_p, \mathbf{z}]\big)\\
&\text{maximize:} \quad \sum_{j,k} r_{jk}x_{jk\mathtt{a}} + \sum_j \bar{r}_j x_{j\mathtt{a}},\\
&\text{subject to:} \quad \sum_{j,k} x_{jk\mathtt{a}} + \sum_j x_{j\mathtt{a}} \le b_h,\\
&\qquad\qquad \sum_j x_{j\mathtt{p}} \le b_p,\\
&\qquad\qquad x_{j\mathtt{a}} + x_{j\mathtt{p}} + x_{j\mathtt{r}} = z_j \qquad \forall\, j \in [n],\\
&\qquad\qquad x_{jk\mathtt{a}} + x_{jk\mathtt{r}} = q_{jk}x_{j\mathtt{p}} \qquad \forall\, j \in [n], k \in [m],\\
&\qquad\qquad \mathbf{x} \ge 0.
\end{aligned}
\tag{7}
$$

Intuitively, $P[b_h, b_p, \mathbf{z}]$ can be understood as follows: given current resource and probing budgets $\mathbf{b}$ and future arrivals $\mathbf{z}$, the decision variables $\mathbf{x} \in \mathbb{R}^{3n+2nm}_{\ge 0}$ represent action summaries, where $x_{j\mathtt{a}}, x_{j\mathtt{r}}, x_{j\mathtt{p}}$ are the total number of future type-j arrivals that are accepted without probing, rejected without probing, and probed, respectively, and $x_{jk\mathtt{a}}, x_{jk\mathtt{r}}$ are the number of probed future type-j arrivals that are revealed to have reward r_{jk}, and then accepted/rejected, respectively. The first two constraints implement the resource budget and probing budget; the third ensures the number of type-j items accepted, probed, or rejected equals arrivals of that type. Finally, the last constraint guarantees that a q_{jk} fraction of probed type-j items have subtype k (i.e., reward r_{jk}).

To construct relaxed value φ, recall that a state is of the form $s = (b_h, b_p, \diamond)$ with $\diamond \in \{\mathtt{a},\mathtt{p},\mathtt{r},\emptyset\}$. For period t (i.e., first stage, $\diamond = \emptyset$), we define $\varphi(t, (b_h, b_p, \emptyset)|\mathcal{G}_t) := P[b_h, b_p, Z^t]$. For $t-1/2$ (i.e., second-stage decisions), we modify φ to incorporate the action $(\mathtt{a},\mathtt{p},\mathtt{r})$ taken in

the first stage. Overall, our relaxation is defined as follows:

$$\varphi(t-1/2,(b_h,b_p,\diamond)|\mathcal{G}_t) = \begin{cases} r_{\xi^{t-1/2}} + P\left[(b_h,b_p),Z^{t-1}\right] & \diamond = \mathtt{a} \\ \max\{r_{\xi^{t-1/2}} + P\left[(b_h-1,b_p),\right. & \\ \quad \left. Z^{t-1}\right], P\left[(b_h,b_p),Z^{t-1}\right]\} & \diamond = \mathtt{p} \\ P\left[(b_h,b_p),Z^{t-1}\right] & \diamond = \mathtt{r}. \end{cases} \tag{8}$$

4.1.3. Value Function Estimate and Online Policy.

Finally, we can use the relaxed value function ϕ in Section 8 to construct an estimated value function $\hat{\varphi}$ by replacing Z^t with $\mathbb{E}_{\xi^{t-1/2}}[Z^t]$. Using this, we get our online policy specified in Algorithm 2.

Algorithm 2 (Probing RABBI)
Input: Access to solutions of $(P[\mathbf{b},\mathbf{z}])$
Output: Sequence of decisions for ONLINE.

1: Initialize budgets $(B_h^T, B_p^T) \leftarrow (B_h, B_p)$
2: **for** period $t = T,\dots,1$ **do**
3: Compute X^t, an optimal solution to $(P[B^t, \mathbb{E}[Z^t]])$.
4: Observe the arrival, say it is of type j, then take action $\hat{U}^t \in \arg\max_{u=\mathtt{a},\mathtt{p},\mathtt{r}}\{X^t_{j,u}\}$.
5: If $\hat{U}^t = \mathtt{r}$ or $\hat{U}^t = \mathtt{a}$: collect zero or random R_j, respectively.
6: If $\hat{U}^t = \mathtt{p}$: probe the arrival to observe $R_j = r_{jk}$, then take action $\arg\max_{u=\mathtt{a},\mathtt{r}}\{X^t_{j,k,u}\}$.
7: Update budgets B^{t-1} accordingly.

Remark 2 (Probing Cost)**.** Our approach can also handle a setting where the controller has no probing budget, but instead incurs a penalty c_j when probing a type-j arrival. The only change to results and proofs is in the definition of $P[\mathbf{b},Z]$, where we drop the constraint involving the probing budget, and modify the objective to be $\max\{\sum_{j,k} r_{jk}x_{j\mathtt{k}\mathtt{a}} + \sum_j \bar{r}_j x_{j\mathtt{a}} - \sum_j c_j x_{j\mathtt{p}}\}$.

4.2. Regret Analysis for Online Probing

We now provide a brief outline of the proof of Theorem 2, which guarantees that Algorithm 2 has a regret that is independent of T, B_h and B_p. Complete proofs are provided in online Appendix EC.2.

The main part of the proof involves showing that φ as defined in Equation (8) obeys the Bellman inequalities (Definition 2) with appropriately chosen Bellman loss. The first ingredient for this is provided by the following lemma, which establishes initial ordering for our relaxed value φ.

Lemma 2. *For any $b_h, b_p \in \mathbb{N}$, and arrivals Z, $\mathbb{E}[V(T,(b_h,b_p)|\mathcal{G}_T)] \le \mathbb{E}[\varphi(T,(b_h,b_p,\emptyset)|\mathcal{G}_T)]$.*

This follows from a standard argument, where we argue that any offline policy induces action summaries that satisfy the constraints defining φ. The proof is provided in online Appendix EC.2.

The bulk of the work is in establishing monotonicity, which we do via the following lemma. Recall the definitions of exclusion sets, satisfying actions and maximum loss (Definitions 3, 4, and 5).

Lemma 3. *Let $\bar{X}$ be a maximizer of $(P[(b_h,b_p),Z^t])$ for some period t, and suppose $\xi^t = i$. Then we have the following implications for satisfying actions:*

1. *If $\bar{X}_{i\mathtt{a}} \ge 1$, then accepting at time t is a satisfying action.*
2. *If $\bar{X}_{i\mathtt{r}} \ge 1$, then rejecting at time t is a satisfying action.*
3. *If $\bar{X}_{i\mathtt{p}} \ge 1$, and $\xi^{t-1/2} = (i,k)$ is such that either $\bar{X}_{ik\mathtt{a}} \ge 1$ or $\bar{X}_{ik\mathtt{r}} \ge 1$, then probing at time t, followed by accepting (if $\bar{X}_{ik\mathtt{a}} \ge 1$) or rejecting (if $\bar{X}_{ik\mathtt{r}} \ge 1$) at time $t-1/2$ is a satisfying action.*

Finally φ satisfies the Bellman inequalities with Bellman loss $L_{\mathcal{B}}(t,(b_h,b_p)) = r_\varphi \mathbb{1}_{\mathcal{B}(t,b_h,b_p)}$, where $\mathcal{B}$ are exclusion sets defined as:

$$\mathcal{B}(t,b_h,b_p) = \{\omega \in \Omega : \nexists \bar{X} \text{ solution to } \left(P\left[(b_h,b_p),Z^t\right]\right) \text{ s.t. (1) } \textbf{\textit{or}} \text{ (2) } \textbf{\textit{or}} \text{ (3) } \textit{hold}\}.$$

The proof generalizes the argument in Example 2 for OnlineKnapsack. We provide a brief outline here, and defer the details to online Appendix EC.2. First, observe that the monotonicity condition in Definition 2 translates to the following condition in the online probing setting:

$$\varphi(t,(b_h,b_p,\emptyset)|\mathcal{G}_t) \le \max_{\diamond\in\{\mathtt{a},\mathtt{p},\mathtt{r}\}}\left\{\mathbb{E}_{\xi^{t-1/2}}\left[\varphi(t-1/2,(s_\diamond,\diamond)|\mathcal{G}_{t-1/2})|\mathcal{G}_t\right]\right\} \quad \forall \omega \notin \mathcal{B}(t,b_h,b_p),$$

where the state $s_\diamond = (b_h-1,b_p)$ if $\diamond = \mathtt{a}$, $s_\diamond = (b_h,b_p-1)$ if $\diamond = \mathtt{p}$, and $s_\diamond = (b_h,b_p)$ if $\diamond = \mathtt{r}$. Moreover, given $\xi^t = i$, we have from Equation (8) that $\mathbb{E}_{\xi^{t-1/2}}[\varphi(t-1/2,(s_\diamond,\diamond))|\mathcal{G}_{t-1/2})|\mathcal{G}_t] = P[(b_h,b_p),Z^{t-1}]$ if $\diamond = \mathtt{r}$, and $r_{\xi^{t-1/2}} + P[(b_h-1,b_p),Z^{t-1}]$ if $\diamond = \mathtt{a}$. Now for cases (1) and (2), the claim in the lemma follows directly by invoking Lemma 1. Finally, case (3) (where $\bar{X}_{i\mathtt{p}} \ge 1$) also follows from using Lemma 1, but in a somewhat more technical way (see online Appendix EC.2 for details).

Using Lemmas 2 and 3, we can complete the regret analysis for Algorithm 2.

Proof of Theorem 2. By Theorem 5, we have that Regret $\le r_\varphi \sum_t (\mathbb{1}_{\mathcal{B}(t,S^t)} + \mathbb{1}_{\mathcal{Q}(t,S^t)})$. To bound this, we proceed in two steps: bounding the measure of the exclusion sets $\mathcal{B}$, and the disagreement sets $\mathcal{Q}$. We conclude using the fact that $r_\varphi \le \max_{j,k} r_{jk}$.

To bound the measure of the exclusion sets $\mathcal{B}$, let $\bar{X}$ be the solution to $(P[\mathbf{b},Z^t])$, and note that Lemma 3 guarantees that there is zero Bellman loss if (1) $\max\{\bar{X}_{j\mathtt{a}},\bar{X}_{j\mathtt{r}}\} \ge 1$, or (2) $\bar{X}_{j\mathtt{p}} \ge 1$ and $\max\{\bar{X}_{jk\mathtt{a}},\bar{X}_{jk\mathtt{r}}\} \ge 1$. The exclusion set $\mathcal{B}(t,\mathbf{b})$ comprises sample paths where both (1) and (2) fail.

Note that any feasible solution to $(P[\mathbf{b}, Z^t])$ satisfies $x_{j\mathtt{a}} + x_{j\mathtt{p}} + x_{j\mathtt{r}} = Z_j^t \ \forall j$ and $x_{jk\mathtt{a}} + x_{jk\mathtt{r}} = q_{jk} x_{j\mathtt{p}} \ \forall j,k$. If $Z_j^t \geq 3$, then one of the variables $x_{j\mathtt{a}}, x_{j\mathtt{p}}, x_{j\mathtt{r}}$ must be at least 1. On the other hand, we need $q_{jk} x_{j\mathtt{p}} \geq 2$ to guarantee that one of $x_{jk\mathtt{a}}, x_{jk\mathtt{r}}$ is at least 1. Thus we have

$$\begin{aligned}\mathbb{P}\left[\mathcal{B}(t,b) | \xi^{t-1/2} = (j,k)\right] &\leq \mathbb{P}\left[Z_j^t < \frac{6}{q_{jk}}\right] \\ &= \mathbb{P}\left[Z_j^t - \mu_j(t) < -\mu_j(t)\left(1 - \frac{6}{\mu_j(t) q_{jk}}\right)\right]. \end{aligned} \tag{9}$$

Restricting $\mu_j(t) \geq 12/q_{jk}$ to ensure the RHS of Equation (9) is positive, we can use a standard Chernoff bound (see Boucheron et al. 2013) to get $\mathbb{P}[\mathcal{B}(t,b) | \xi^{t-1/2} = (j,k)] \leq e^{-2(p_j/2)t} + \mathbb{1}_{\{t \leq 12/(p_j q_{jk})\}}$. Finally,

$$\begin{aligned}\sum_t \mathbb{P}\left[\mathcal{B}(t, B^t)\right] &\leq \sum_t \sum_j p_j e^{-2(p_j/2)t} \\ &\quad + \sum_t \sum_{j,k} p_j q_{jk} \mathbb{1}_{\{t \leq 12/(p_j q_{jk})\}} \\ &\leq \sum_j \frac{2}{p_j} + 12.\end{aligned}$$

To bound the information loss $\sum_t \mathbb{P}[\mathcal{Q}(t, S^t)]$, recall $\mathcal{Q}(t, S^t) \subseteq \Omega$ is the event where $\hat{U}^t$ is not satisfying. Let $\bar{X}$ be a solution to $(P[\mathbf{b}, Z^t])$, t a first stage, and let $j = \xi^t$. We now have two cases depending on if $\hat{U}^t \in \{\mathtt{a}, \mathtt{r}\}$ or $\hat{U}^t = \mathtt{p}$. First, if $\hat{U}^t \in \{\mathtt{a}, \mathtt{r}\}$, then according to Lemma 3, accepting or rejecting is satisfying whenever $\max\{\bar{X}_{j\mathtt{a}}, \bar{X}_{j\mathtt{r}}\} \geq 1$. Since $X^t(\xi^t, \hat{U}^t) = \max\{X^t(\xi^t, u) :\}$ $u = \mathtt{a}, \mathtt{p}, \mathtt{r}$ and $X_{j\mathtt{a}}^t + X_{j\mathtt{p}}^t + X_{j\mathtt{r}}^t = \mu_j(t)$, we have

$$\begin{aligned}&\mathbb{P}\left[\bar{X}(j, \hat{U}^t) < 1 | X^t(j, \hat{U}^t) \geq \mu_j(t)/3\right] \\ &\quad \leq \mathbb{P}\left[\left\|\bar{X} - X^t\right\|_\infty \geq \mu_j(t)/3\right].\end{aligned}$$

On the other hand, if $\hat{U}^t = \mathtt{p}$, the error is bounded by

$$\begin{aligned}&\mathbb{P}\left[\bar{X}_{j\mathtt{p}} < 1 \text{ or } \bar{X}_{\xi^{t-1/2},u} < 1 \middle| X_{j\mathtt{p}}^t \geq \frac{\mu_j(t)}{3}, X_{\xi^{t-1/2},u}^t \right. \\ &\left. \geq \frac{q_{\xi^{t-1/2}} \mu_j(t)}{6}\right] \leq \mathbb{P}\left[\left\|\bar{X} - X^t\right\|_\infty \geq \frac{q_{\xi^{t-1/2}} \mu_j(t)}{6}\right],\end{aligned}$$

where u is the action with largest value between the variables $X^t(\xi^{t-1/2}, \mathtt{a}), X^t(\xi^{t-1/2}, \mathtt{r})$.

Thus, regardless of the action $\hat{U}^t$, the probability of choosing a nonsatisfying action is bounded by $\mathbb{P}[\|\bar{X} - X^t\|_\infty \geq \min_k q_{jk} \cdot \mu_j(t)/6]$. Moreover, standard LP sensitivity results (Mangasarian and Shiau 1987, theorem 2.4) imply that there exists κ depending on $\mathbf{q}, n, m$ alone, such that $\|\bar{X} - X^t\|_\infty \leq \kappa \|Z^t - \mu(t)\|_1$. Finally, the measure of sets $\mathcal{Q}$ where Online chooses a nonsatisfying action is bounded by

$$\begin{aligned}&\sum_t \mathbb{P}\left[\mathcal{Q}(t, S^t)\right] \\ &\quad \leq \sum_t \mathbb{P}\left[\left\|Z^t - \mu(t)\right\|_1 \geq \min_k q_{jk} \cdot \mu_j(t)/6\kappa\right] < \infty.\end{aligned}$$

The summability follows arguments presented in (Vera and Banerjee 2020), based on standard concentration bounds.

5. Dynamic Pricing

We now apply our framework to dynamic pricing. In the basic setting, we have d resources and n customer types. Each customer type has a private reward for a set of resources. The controller observes the customer type, and if the corresponding set of resources is available, posts a price. The customer then purchases iff the requested set is available and the posted price is below the private reward. The resource consumption is encoded in a matrix $A \in \{0,1\}^{d \times n}$. In Section 5.5, we generalize to settings where rather than requesting a specific set of products, customers choose between multiple substitute bundles of resources.

We consider the following formal model: at time t, type $j \in [n]$ arrives with probability p_j, is seen by the controller, who then posts a price f_{jl} from a set of available prices $\{f_{j1}, \ldots, f_{jm}\}$. The customer then draws a private reward $R^t \sim F_j$, and a purchase occurs iff $R^t > f_{jl}$. If the customer buys, f_{jl} is collected and the inventory decreases by A_j. On the other hand, if the customer does not buy, the controller collects zero and the inventory remains unchanged.

5.1. Offline Benchmark and Online Policy for Dynamic Pricing

5.1.1. Offline Benchmark. Note that for each customer type j, there are Z_j^T arrivals, and hence Z_j^T draws from the distribution F_j. We now define our benchmark by considering Offline to be a controller that knows the realized histogram of these draws, that is, for each j, Offline knows the empirical distribution of the Z_j^T rewards. Moreover, at the end of each period t, Offline also observes the realized valuation R^t whether there is a sale. Note that Offline does not know the exact sequence of these rewards, and so is not a full-information benchmark. For example, say $Z_1^T = 15$ and we reveal that 10 arrivals type-1 have private reward \$1 and five arrivals type-1 have private reward \$2. Now, upon observing a type-1 arrival, Offline concludes that the reward is \$2 with probability $\frac{5}{15}$. Now if the arrival had value \$1, then, the next time Offline observes a type-1, its belief is that the reward is \$2 with probability $\frac{5}{14}$.

Formally, for each j, suppose the prices are ordered $f_{j1} > f_{j2} > \ldots > f_{jm}$. Denote $\xi^t \in [n]$ to be the type of the arrival at time t. To define Offline, we introduce a sequence of independent random vectors $\{Y^t : t = T, T-1, \ldots, 1\}$ where $Y_{jl}^t := \mathbb{1}_{\{\xi^t = j, R^t > f_{jl}\}}$; in other words, Y_{jl}^t is the indicator of whether a price f_{jl} or lower is accepted by the type-j at time t. We define $Q_{jl}(t) := \frac{1}{Z_j^t} \sum_{\tau=1}^t Y_{jl}^\tau$ to be the fraction of type-j

customers who accept price f_{jl} in the last t periods. Observe that $Q_{jl}(t)$ is a martingale with $\mathbb{E}[Q_{jl}(t)] = \bar{F}_j(f_{jl})$ and $Q_{jl}(t) = \frac{Z_j^{t+1}}{Z_j^t} Q_{jl}(t+1) - \frac{1}{Z_j^t} Y_{jl}^{t+1}$.

Offline's information is now given by the filtration $\mathcal{G}_t = \sigma(\{Q(\tau), Z^\tau : \tau \geq t\})$, that is, at every time t, Offline knows the total demand Z_j^t and the empirical averages $Q_{jl}(t)$, but not the sequence of rewards. This coincides with the canonical filtration (Definition 1) with variables $(Q_{jl}(T), Z_j^T : j \in [n], l \in [m])$. The filtration $\mathcal{G}$ is strictly coarser than the full-information filtration, which would correspond to revealing all the variables $Y^T, Y^{T-1}, \ldots, Y^1$ instead of their empirical averages.

5.1.2. Relaxed Value Function.

Consider the following LP, parameterized by $(\mathbf{b}, \mathbf{q}, \mathbf{z})$:

$$\begin{aligned} (P[\mathbf{b}, \mathbf{q}, \mathbf{z}]) \quad \text{maximize:} \quad & \sum_{j,l} f_{jl} q_{jl} x_{jl} \\ \text{subject to:} \quad & \sum_{j,l} a_{ij} q_{jl} x_{jl} \leq b_i \quad \forall\, i \in [d] \\ & \sum_{j,l} x_{jl} + x_{j\mathtt{r}} = z_j \quad \forall\, j \in [n] \\ & \mathbf{x} \geq 0 \end{aligned} \tag{10}$$

We define the relaxed value as $\varphi(t, \mathbf{b}|\mathcal{G}_t) := P[\mathbf{b}, Q(t), Z^t]$, and the corresponding estimated value as $\hat{\varphi}(t, \mathbf{b}) := P[\mathbf{b}, \mathbf{q}, t\mathbf{p}]$, where $q_{jl} = \bar{F}_j(f_{jl})$. The resulting rabbi policy is presented in Algorithm 3.

Algorithm 3 (Pricing rabbi)
Input: Access to solutions of $(P[\mathbf{b}, \mathbf{q}, \mathbf{z}])$
Output: Sequence of decisions for Online.

1: Set $B^T \leftarrow B$ as the given initial budget and $q_{jl} \leftarrow \bar{F}_j(f_{jl})$
2: **for** $t = T, \ldots, 1$ **do**
3: If the arrival is type j and $A_j \not\leq B^t$: not enough resources, reject and go to $t-1$.
4: Compute X^t, an optimal solution to $(P[B^t, \mathbf{q}, t\mathbf{p}])$.
5: Let $l \in \arg\max\{X_{j,l}^t : l = 1, \ldots, m, \mathtt{r}\}$. If $l = \mathtt{r}$, reject and go to $t-1$. Else post price f_{jl}.
6: If $R^t > f_{jl}$, collect f_{jl} and $B^{t-1} \leftarrow B^t - A_j$; else $B^{t-1} \leftarrow B^t$.

To get some intuition into the LP $(P[\mathbf{b}, \mathbf{q}, \mathbf{z}])$, note that if $q_{jl} = \bar{F}_j(f_{jl})$, that is, the probability that price f_{jl} is accepted by a type-j customer and z_j is the number of type-j arrivals, then $(P[\mathbf{b}, \mathbf{q}, \mathbf{z}])$ can be interpreted as follows: the variable x_{jl} represents the number of times that price f_{jl} is offered, with $\sum_{j,l} f_{jl} q_{jl} x_{jl}$ the expected reward from the corresponding arrivals. Each time price f_{jl} is offered, $a_{ij} q_{jl}$ units of resource i are consumed in expectation, and hence $\sum_{j,l} a_{ij} q_{jl} x_{jl}$ is the total expected consumption of resource i. Finally, at most one price is offered per arrival, which is captured by $\sum_l x_{jl} + x_{j\mathtt{r}} = z_j$, where $x_{j\mathtt{r}}$ is the number of rejected type-j customers.

5.2. Bellman Inequalities and Bellman Loss

We first argue that our choice of φ satisfies the Bellman inequalities.

Lemma 4. *Let $V(T, B|\mathcal{G}_T)$ be the value of* Offline*'s optimal policy, and $\varphi(t, \mathbf{b}|\mathcal{G}_t) = P[\mathbf{b}, Q(t), Z^t]$ be the relaxed value with optimal solution X:*

1. The relaxed value satisfies the initial ordering condition: $\mathbb{E}[V(T, B|\mathcal{G}_T)] \leq \mathbb{E}[\varphi(T, B|\mathcal{G}_T)]$.
2. *If the arriving type is j and $\max_l\{X_{jl}\} \geq 1$, then $\mathbb{E}[L_B(t, \mathbf{b})] \leq 0$.*
3. *If the arriving type is j and $X_{jl} \geq 1$, then posting f_{jl} is a satisfying action.*

We omit the proof of the initial ordering in item (1), as it is similar to that of Lemma 2. Next we present the main ingredients for obtaining the monotonicity property (items (2) and (3)). Complete details are deferred to online Appendix EC.3. For ease of exposition, when the controller rejects, the controller can equivalently post $f_{j\mathtt{r}} = \infty$ such that $\bar{F}_j(f_{j\mathtt{r}}) = 0$ with the convention $0 \times \infty = 0$.

We start by recalling the monotonicity condition (Definition 2). Denote $\mathbb{E}_t[\cdot] = \mathbb{E}[\cdot|\mathcal{G}_t]$. If the inventory is $\mathbf{b} \geq A_j$, the random reward of posting price f_{jl} at t is $f_{jl} Y_{jl}^t$ and the random new inventory is $\mathbf{b} - A_j Y_{jl}^t$, thus monotonicity corresponds to:

$$\begin{aligned} \varphi(t+1, \mathbf{b}) \leq \max_{l \in [m] \cup \{\mathtt{r}\}} \left\{ \mathbb{E}_{t+1}\left[f_{jl} Y_{jl}^{t+1} + \varphi\left(t, \mathbf{b} - A_j Y_{jl}^{t+1}\right) \right] \right\} \\ + \mathbb{E}_{t+1}[L_B(t+1, \mathbf{b})]. \end{aligned}$$

Because Q is a martingale, we have $\mathbb{E}_t[Y^t] = Q(t)$, and we can further simplify the condition to:

$$\begin{aligned} \varphi(t+1, \mathbf{b}) \leq \max_{l \in [m] \cup \{\mathtt{r}\}} \left\{ f_{jl} Q_{jl}(t+1) + \mathbb{E}_{t+1}\left[\varphi\left(t, \mathbf{b} - A_j Y_{jl}^{t+1}\right) \right] \right\} \\ + \mathbb{E}_{t+1}[L_B(t+1, b)]. \end{aligned} \tag{11}$$

Define $L_B(t+1, b, j, l) := \varphi(t+1, \mathbf{b}) - f_{jl} Q_{jl}(t+1) - \mathbb{E}_{t+1}[\varphi(t, \mathbf{b} - A_j Y_{jl}^{t+1})]$, which corresponds to the loss in Equation (11) when we assume a specific price f_{jl} is posted. Recall we define $\varphi(t+1, \mathbf{b}) = P[\mathbf{b}, Q(t+1), Z^{t+1}]$. Moreover, for an arrival of type j and any solution X of $P[\mathbf{b}, Q(t+1), Z^{t+1}]$, if $X_{jl} \geq 1$, then using Lemma 1, we have $P[\mathbf{b}, Q(t+1), Z^{t+1}] = f_{jl} Q_{jl}(t+1) + P[\mathbf{b} - A_j Q_{jl}(t+1), Q(t+1), Z^t]$. Thus, assuming $X_{jl} \geq 1$, we can write the loss in the Bellman inequality as

$$\begin{aligned} L_B(t+1, \mathbf{b}, j, l) = P\left[\mathbf{b} - A_j Q_{jl}(t+1), Q(t+1), Z^t\right] \\ - \mathbb{E}_{t+1}\left[P\left[\mathbf{b} - A_j Y_{jl}^{t+1}, Q(t), Z^t\right] \right]. \end{aligned} \tag{12}$$

Observe that $L_B(t,\mathbf{b},j,l)$ is characterized by a random LP that depends on Y^{t+1} (which is unknown at time $t+1$), see Equation (12). To complete item (2) of Lemma 4, it remains to prove that $L_B(t,\mathbf{b},j,l)$ characterized in (12) satisfies $\mathbb{E}_t[L_B(t,\mathbf{b},j,l)] \le 0$. This is proved in online Appendix EC.3 by arguing that the term in (12) is upper bounded by a zero-mean random variable.

We can then conclude that, for each l with $X_{jl} \ge 1$, $\mathbb{E}_{t+1}[L_B(t+1,b,j,l)] \le 0$, so that $\varphi(t+1,\mathbf{b}) \le \mathbb{E}_{t+1}[f_{jl}Q_{jl}(t+1) + \varphi(t,\mathbf{b}-A_jY_{jl}^{t+1})]$, implying that posting price f_{jl} is a satisfying action, which is item (3) of Lemma 4.

5.3. Information Loss and Overall Performance Guarantee

Next we study the disagreement sets $\mathcal{Q}(t,B^t)$ and bound the information loss $\sum_t \mathbb{P}[\mathcal{Q}(t,B^t)]$.

Proposition 2. *Let X be a solution of $(P[\mathbf{b},Q(t),Z^t])$. If $X_{jl} \ge 1$, then posting f_{jl} is a satisfying action. Furthermore, the information loss is bounded by $\mathbb{P}[\mathcal{Q}(t,B^t)] \le 1/t^2$ for all $t \ge c$, where c depends only on $(\mathbf{f},\mathbf{p},A,F_1,\ldots,F_n)$.*

We now give an outline of this proof (for details, refer to online Appendix EC.3). Recall that RABBI chooses l as the maximum entry of the solution to $(P[\mathbf{b},\mathbb{E}[Q(t)],\mathbb{E}[Z^t]])$, which is a perturbed version of the object of interest, thus ONLINE needs to guess l such that $X_{jl} \ge 1$ without the knowledge of $Q(t)$ and Z^t, creating an information loss.

To build intuition, consider the case where $d=1$ and $n=1$, that is, selling multiple copies of an item to homogeneous customers. Since there is only one type, we drop the index j. Recall $f_1 > \ldots > f_m$ and $q_1 < \ldots < q_m$. It is easy to check that the solution of $P[b,\mathbf{q},t]$ is as follows: (i) if $b \le tq_1$, then $x = (b/q_1, 0,\ldots,0)$; (ii) if $b > tq_m$, then $x = (0,\ldots,0,t)$; (iii) otherwise, if $b \in (tq_l, tq_{l+1}]$, then $x_{l'} = 0$ for $l' \ne l, l+1$, and $x_l = (tq_{l+1}-b)(q_{l+1}-q_l), x_{l+1} = (b-tq_l)(q_{l+1}-q_l)$. Figure 4 illustrates this solution, and also shows that for RABBI's guess to be incorrect, $Q(t)$ and $\mathbb{E}[Q(t)]$ must deviate considerably, which the next lemma indicates is unlikely. This intuition carries over to higher dimensions.

Lemma 5. *For any $j \in [n]$, there is a constant c_j depending on p_j only such that, for any time t, $\mathbb{P}[\max_l Q_{jl}(t) - \mathbb{E}[Q_{jl}(t)] > \sqrt{\frac{\log(t)}{t}}] \le \frac{c_j}{t^2}$.*

Proof. From the DKW inequality (Massart 1990) for empirical measures, we have

$$\mathbb{P}\left[\sup_l Q_{jl}(t) - \bar{F}(f_{jl}) > \lambda \middle| Z^t\right] \le 2e^{-2\lambda^2 Z_j^t}.$$

Also for $Z_j^t \sim \mathrm{Bin}(t,p_j)$, $\mathbb{E}[e^{-\theta Z_j^t}] = (1-p+pe^{-\theta})^t$. Setting $\lambda = \sqrt{\log(t)/t}$, we get

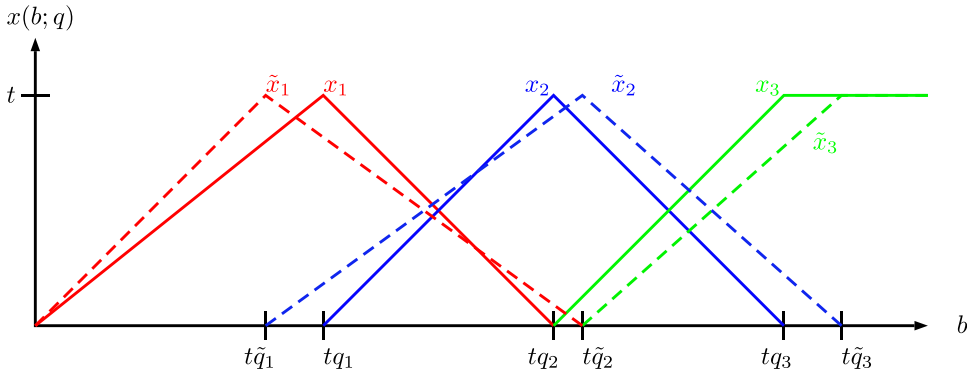
$$\mathbb{P}\left[\sup_l Q_{jl}(t) - \bar{F}(f_{jl}) > \sqrt{\frac{\log(t)}{t}}\right] \le 2\left(1-p_j+p_je^{-\theta}\right)^t \quad \text{where } \theta = 2\log(t)/t.$$

Using the inequality $e^{-\theta} \le 1-\theta+\theta^2/2$, an algebraic check confirms the desired inequality. □

5.3.1. Stability of Left-Hand Side Perturbations.

As stated in Algorithm 3, ONLINE takes actions based on $P[\mathbf{b},\mathbb{E}[Q(t)],\mathbb{E}[Z^t]]$, whereas OFFLINE uses $P[\mathbf{b},Q(t),Z^t]$. Therefore, for fixed $(t,\mathbf{b})$, we need to compare solutions of $P[\mathbf{b},\mathbf{q},\mathbf{z}]$ to those of $P[\mathbf{b},\mathbf{q}+\Delta\mathbf{q},\mathbf{z}+\Delta\mathbf{z}]$, where Δ is the perturbation. Define $\mathbf{q} = \mathbb{E}[Q(t)]$, $\mathbf{z} = \mathbb{E}[Z^t]$, $\Delta\mathbf{q} = Q(t) - \mathbb{E}[Q(t)]$, and $\Delta\mathbf{z} = Z^t - \mathbb{E}[Z^t]$.

Figure 4. (Color online) Solution to the Pricing LP in Equation (10) for the Case $d=1$ and $n=1$, Which Correspond to Selling Multiple Copies of an Item to Homogeneous Customers



Notes. If $b/t \in (q_l, q_{l+1}]$, the prices used by the LP are f_l, f_{l+1} and the amount of time we offer each is piecewise linear in the budget. For a perturbation $\tilde{\mathbf{q}}$ of $\mathbf{q}$, we superpose the solutions with the different parameters. Our guess is incorrect only when $\tilde{x}_l \gg 1$ and $x_l < 1$, which necessitates a substantial perturbation of $\mathbf{q}$.

Lemma 6 (Selection Program). *Let* $V_t = P[\mathbf{b}, \mathbf{q} + \Delta\mathbf{q}, \mathbf{z} + \Delta\mathbf{z}]$ *and fix a component* (j', l'). *Then posting price* $f_{j'l'}$ *is satisfying if* $P_S[V_t, \mathbf{q} + \Delta\mathbf{q}, \mathbf{z} + \Delta\mathbf{z}] \geq 1$, *where*

$$P_S\big[V_t, \mathbf{q} + \Delta\mathbf{q}, \mathbf{z} + \Delta\mathbf{z}\big] := \max\left\{ x_{j'l'} : \sum_{j,l} f_{jl}(q_{jl} + \Delta q_{jl}) x_{jl} \geq V_t, \mathbf{x} \text{ feasible for } P\big[\mathbf{b}, \mathbf{q} + \Delta\mathbf{q}, \mathbf{z} + \Delta\mathbf{z}\big] \right\}.$$

In other words, $\mathcal{Q}(t, b, l) = \{\omega \in \Omega : P_S[V_t[\omega], Q(t), Z^t] < 1\}$.

Proof. This problem selects, among all the solutions of $P[\mathbf{b}, \mathbf{q} + \Delta\mathbf{q}, \mathbf{z} + \Delta\mathbf{z}]$, one with the largest component $X_{j'l'}$. From Lemma 4, we know that if $X_{j'l'} \geq 1$, then posting $f_{j'l'}$ is satisfying. □

We have converted the condition "$\exists X$ solving $P[v, \mathbf{q} + \Delta\mathbf{q}, \mathbf{z} + \Delta\mathbf{z}]$ with $X_{j'l'} \geq 1$" to an optimization program. Let $\bar{\mathbf{x}}$ be the solution to the proxy $P[\mathbf{b}, \mathbf{q}, \mathbf{z}]$ and let v_t be the objective value (recall that V_t is the value of $P[\mathbf{b}, \mathbf{q} + \Delta\mathbf{q}, \mathbf{z} + \Delta\mathbf{z}]$). Since the algorithm picks the price with the largest component, assume $\bar{x}_{j'l'} = \max_l \bar{x}_{j'l} \gg 1$. In particular, $P_S[v_t, \mathbf{q}, \mathbf{z}] \gg 1$ for this fixed (j', l'). We want to show that $P_S[V_t, \mathbf{q} + \Delta\mathbf{q}, \mathbf{z} + \Delta\mathbf{z}] \geq 1$ for that particular (j', l'). To that end, we need to bound the difference between $P_S[V_t, \mathbf{q} + \Delta\mathbf{q}, \mathbf{z} + \Delta\mathbf{z}]$ and $P_S[v_t, \mathbf{q}, \mathbf{z}]$. This difference depends on (i) $v_t - V_t$, (ii) Δ, and (iii) the dual variables of $(P_S[V_t, \mathbf{q} + \Delta\mathbf{q}, \mathbf{z} + \Delta\mathbf{z}])$. Observe that the quantities (i)–(iii) are random. We state the result next. The proof is provided in online Appendix EC.3.

Lemma 7. *There is a constant* c *that depends only on* $(\mathbf{f}, \mathbf{p}, A, F_1, \ldots, F_n)$ *such that, for all* $t \geq c$, *with probability* $1 - c/t^2$, $P_S[V_t, Q(t), Z^t] - P_S[v_t, \mathbb{E}[Q(t)], \mathbb{E}[Z^t]] \geq -c\sqrt{t \log(t)}$.

Lemma 7 leads to the bound in Proposition 2. Indeed, since the LP in Equation (10) has the constraint $\sum_{l \in [m] \cup \{\mathrm{r}\}} \bar{x}_{jl} = tp_j$, the maximum entry is guaranteed to have a value of at least $tp_j/(m+1)$. Therefore, by definition of the selection program, $P_S[v_t, \mathbb{E}[Q(t)], \mathbb{E}[Z^t]] \geq tp_j/(m+1)$. We know that posting $f_{jl'}$ is satisfying whenever $P_S[V_t, Q(t), Z^t] \geq 1$ (see Lemma 6), hence posting the maximum entry is satisfying provided that $tp_j/(m+1) - c\sqrt{t \log(t)} \geq 1$, which holds for all t large enough.

5.4. Numerical Simulations

We test our algorithm on two systems, henceforth the small system and the large system. For each system, we consider a sequence of instances with increasing horizons and initial inventories.

The small system corresponds to the one-dimensional problem ($n = 1$ and $d = 1$). In this case, we can solve the DP for small enough horizons and directly compute the optimality gap. The large system corresponds to a multidimensional problem with $n = 20$, $d = 25$, and $m = 3$. The DP solution is intractable for the large system, yet we can compute the offline benchmark and compare our algorithm against it. The optimality gap, recall, is bounded by the offline versus RABBI gap.

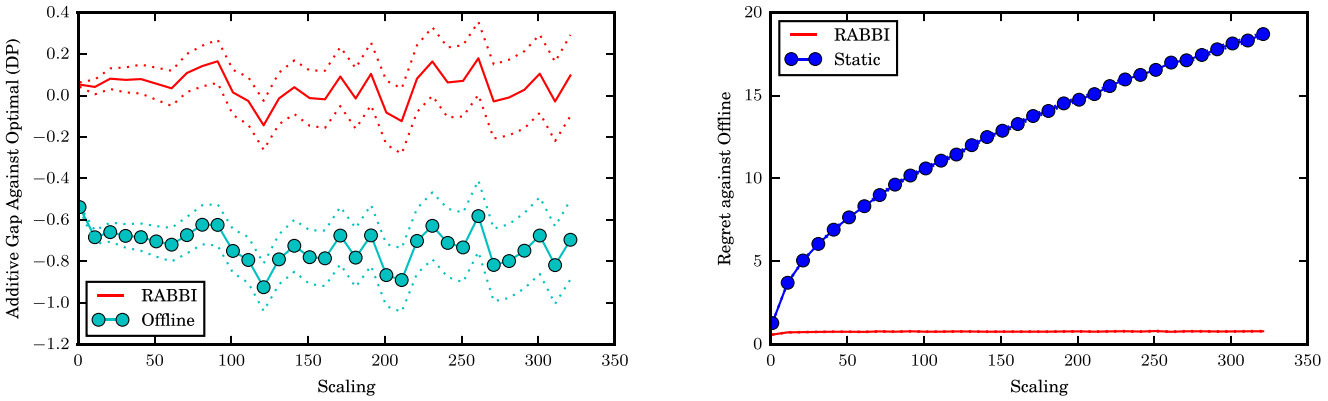
For the small system, the kth instance has budget $B = 6k$ and horizon $T = 20k$. For each scaling k, we run 100,000 simulations. We consider the following primitives: prices are $(1, 2, 3)$ and the private reward R^t has an atomic distribution on $(1, 2, 3)$ with probabilities $(0.3, 0.4, 0.3)$. The instance is chosen such that it is dual degenerate for (10), which is supposedly the more difficult case (Jasin 2014). For the large system, the parameters were generated randomly and are reported in online Appendix EC.6. The kth instance has horizon $T = 100k$ and budgets $B_i = 10k$ for all $i \in [25]$.

For the small system, we consider k small enough (short horizon) so that we can compute the optimal policy. This computation becomes intractable already for moderate values of k (RABBI however scales gracefully with k as it only requires resolving an LP in each period). In Figure 5 (left) we display the gap between the optimal solution and both the RABBI and OFFLINE's value. We make two observations: (i) the OFFLINE benchmark outperforms the optimal (as it should), but by a rather small margin; and (ii) RABBI has a constant regret (i.e., independent of k) relative to OFFLINE, and hence constant optimality gap. In contrast, a full-information benchmark would outperform the optimal by too much to be useful.

In Figure 5 (right), we compare RABBI to the optimal static pricing policy, which has regret $\Omega(\sqrt{k})$ (Gallego and Van Ryzin 1997). In particular, if $D(f)$ denotes the demand at fare f, we choose the static price to be the one that maximizes the revenue function $f \cdot D(f) = f \cdot T \cdot \bar{F}(f)$, subject to the constraint $D(f) \leq B$. The solution is the better of two prices: (i) the market clearing price, that is, that satisfies $D(f) = B$, or (ii) the monopoly price, which maximizes $fD(f)$. We note though that when a continuum of prices is allowed, Jasin (2014) proposes an algorithm (which, like RABBI, is based on resolving an optimization problem in each period) that achieves a regret that is logarithmic in k under certain nondegeneracy assumptions on the optimization problem and differentiability assumptions on the valuation distribution. In contrast, our constant-regret guarantees hold under a finite price menu.

In Figure 6, we display the results for the large system. Here, since the DP is intractable, we use the offline benchmark. The resulting regret is negligible relative to the total value, as captured by the approximation factor on the right-hand side of the figure. We also present the competitive ratio of OFFLINE against the full-information benchmark (this upper bounds the

Figure 5. (Color online) Regret in the Small System ($n = 1$ and $d = 1$), with Horizon $T = 20k$ and Initial Budget $B = 6k$, Under Scaling $k = 1, 10, 20, \ldots, 340$



Notes. Dotted lines represent 90% confidence interval. (Left) additive gaps against the optimal policy, i.e., $V^{DP} - V^{\text{RABBI}}$ and $V^{DP} - V^{\text{OFFLINE}}$. (Right) Regret of two policies against OFFLINE.

competitive ratio of any nonanticipatory policy) and observe that is bounded away from 1, hence showing that the full-information benchmark is $\Omega(T)$ away from the DP in our randomly generated instance, which confirms the need for our refined benchmark.

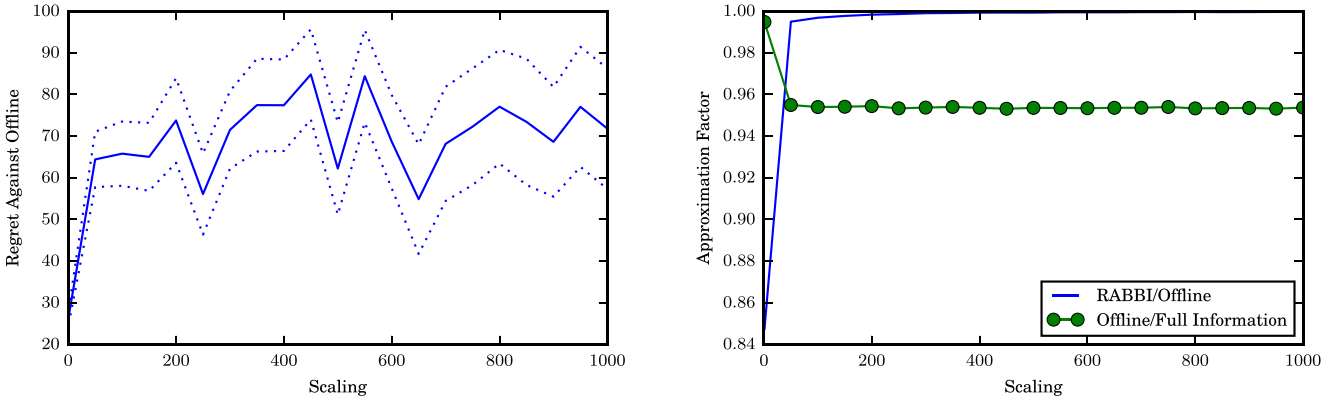
5.5. Posted Pricing with Customer Choice

We now consider settings where customers, rather than requesting a specific product, choose between multiple substitutes. As a concrete example, consider a hardware store selling washers and dryers. The store can set a separate price for a washer, a dryer, and also for buying a washer-and-dryer bundle (i.e., one of each). An incoming customer sees the prices and chooses to buy each of the three options (or nothing at all) with some probability depending on the price menu. See Talluri and Van Ryzin (2006, chapter 7) for details on such customer-choice models. For exposition, we focus here on a single-customer type, with arbitrary (but known) customer-choice model.

As before, the controller chooses a price to post for each product and selling one unit of product $j \in [n]$ depletes resources according to $A_j \in \{0,1\}^d$. There is a discrete set of assortment menus, denoted by $\mathcal{A}$. An assortment $\alpha \in \mathcal{A}$ is associated with a vector of prices $(f_{1\alpha}, \ldots, f_{n\alpha})$, one price per product. Setting $f_{j\alpha} = \infty$ corresponds to not offering product j. Note that if each product's price is restricted to take one of m distinct values, then there are at most $\mathcal{A} \le m^n$ different assortments. The actual number of relevant assortments might, however, be much smaller than this.

An arriving customer, when offered assortment α, chooses to buy product j with a probability $p_j(\alpha)$, with $\sum_{j=0}^{n} p_j(\alpha) = 1$ (where we use $j = 0$ for the no-purchase option). These probabilities might be derived, for example, from a standard family such as

Figure 6. (Color online) Performance in the Large System ($n = 20$ and $d = 25$) with Horizon $T = 100k$ and Initial Budgets $B_i = 10k$ for $i \in [25]$, Under Scaling $k = 1, 2, \ldots, 1{,}000$



Notes. (Left) Regret against OFFLINE, that is, $V^{\text{OFFLINE}} - V^{\text{RABBI}}$; dotted lines represent 90% CI. (Right) Approximation factors $V^{\text{RABBI}}/V^{\text{OFFLINE}}$ of RABBI compared with OFFLINE, and $V^{\text{OFFLINE}}/V^{\text{Full-Info}}$ of OFFLINE against the full-information benchmark. This shows that the full-information benchmark is indeed too loose, as it is $\Omega(T)$ away from the DP.

the multinomial-logit model, nested logit model, and so on; our results do not need any specific structure on the choice probabilities (although assuming more structure may lead to better regret scaling with respect to the number of price menus and more efficient ways of solving the resulting LP relaxation).

The process unfolds as follows: (i) at time t, the controller posts an assortment $\alpha \in \mathcal{A}$; (ii) with probability $p_j(\alpha)$, the arriving customer buys one unit of product j (with product 0 corresponding to no purchase). Now given the choice probabilities, we can simulate the choice model as follows: we assume w.l.o.g. that the customer arriving at time t is endowed with an i.i.d. random variable $\xi^t \sim \text{Uniform}(0,1)$, and assert that the customer buys product j if $\xi^t \in [\sum_{j'=0}^{j-1} p_{j'}(\alpha), \sum_{j'=0}^{j} p_{j'}(\alpha)]$. Note that the order of products here is arbitrary.

Applying RABBI to this setting gives the following result.

Theorem 6 (Dynamic Pricing with Customer Choice)**.** *For any choice model with probabilities and prices* $(p_j(\alpha), f_{j\alpha} : j \in [n], \alpha \in \mathcal{A})$, RABBI *obtains a regret that depends only on* (A, p, f)*, but is independent of the horizon length* T *and initial budget levels* $B \in \mathbb{N}^d$.

5.5.1. Algorithm and Analysis.

The following LP extends Equation (10) to incorporate consumer choice:

$$\begin{aligned}(P[\mathbf{b},\mathbf{q},\mathbf{z}]) \quad \text{maximize:} \quad & \sum_{\alpha\in\mathcal{A}} x_\alpha \sum_{j\in[n]} f_{j\alpha} q_{j\alpha} \\ \text{subject to:} \quad & \sum_{\alpha\in\mathcal{A}} \sum_{j\in[n]} a_{ij} q_{j\alpha} x_\alpha \le b_i \quad \forall\, i \in [d], \\ & \sum_{\alpha\in\mathcal{A}} x_\alpha = t \\ & \mathbf{x} \ge 0. \end{aligned} \tag{13}$$

Here, $q_{j\alpha}$ stands for the fraction of customers that would buy product j if presented with the price assortment α. RABBI re-solves, in each period, this LP with the expected fraction $q_{j\alpha} = p_j(\alpha)$. In contrast, OFFLINE knows $Q_{j\alpha}(t)$, the realized fraction of customers that, given assortment α, would buy product j (formally, OFFLINE is equipped with the canonical augmented filtration with variables $(Q_{j\alpha}(T) : j \in [n], \alpha \in \mathcal{A})$), and solves Equation (13) with $q_{j\alpha} = Q_{j\alpha}(t)$, where:

$$Q_{j\alpha}(t) := \frac{1}{t}\sum_{\tau=1}^{t} Y_{j\alpha}^t$$
$$\text{where } Y_{j\alpha}^t := \mathbb{1}_{\{\sum_{j'=0}^{j-1} p_{j'}(\alpha) \le \xi^t \le \sum_{j'=0}^{j} p_{j'}(\alpha)\}}.$$

With the (re)defined key ingredients—namely the LP in Equation (13) and OFFLINE's information structure—it is evident that that the analysis of this expanded model is identical to that of the basic (no-choice) pricing setting with obvious changes. For example, if assortment α is posted at time t, the random collected reward is $\sum_j Y_{j\alpha}^t f_{j\alpha}$ and the random inventory at $t-1$ is $b - \sum_j A_j Y_{j\alpha}^t$. In turn, the Bellman loss in Equation (12) takes on the following form:

$$L_B(t+1,\mathbf{b},\alpha) = P\left[\mathbf{b} - \sum_j A_j Q_{j\alpha}(t+1), Q(t+1), t\right] - \mathbb{E}_{t+1}\left[P\left[\mathbf{b} - \sum_j A_j Y_{j\alpha}^{t+1}, Q(t), t\right]\right]. \tag{14}$$

Now we have a sum over products j, but the analysis goes through via linearity of expectations.

5.5.2. Numerical Simulations.

We demonstrate our algorithm for the following simple choice model with two resources $(R1, R2)$, and three products $(\{R1\}, \{R2\}, \{R1, R2\})$ (for example, a hardware store selling washers $(R1)$, dryers $(R2)$, or washer-and-dryer combos $\{R1, R2\}$). The controller has initial inventories of each resource, and can choose among one of seven price assortments: high and low prices with/without discounts for buying the bundle, and price menus assuming stock-out of either or both resource. The price menus and choice probabilities are detailed in Table 1. We run RABBI for this instance while scaling the horizon and initial inventory (see Figure 7).

6. Online Knapsack with Distribution Learning

Finally, we consider the distribution-agnostic online knapsack setting. We study first the full feedback setting, and in Section 6.3 extend to censored feedback. As in the baseline OnlineKnapsack, at each time t, the arrival is of type $j \in [n]$ with known probability p_j. Type j has a known weight w_j and random reward R_j, drawn from a distribution F_j, with $r_j := \mathbb{E}[R_j]$. Critically, we assume r_j and F_j are unknown to ONLINE.

The reward R_j is revealed only after the decision to accept/reject has been made. At the end of each period, we observe the realization of both accepted and rejected items. In contrast, OFFLINE has access to the distribution F_j, but not to the realizations. We assume that, before the process starts, we are given one sample of each type, and with t periods to go, define R_j^t to be the empirical average of the observed rewards for type-j arrivals.

As in probing, we divide each period $t \in \{T, T-1, \ldots, 1\}$ into two stages, t and $t-1/2$. In the first stage (i.e., period t), the input reveals the type $j \in [n]$, and in second stages (i.e., period $t-1/2$), the reward is revealed. The random inputs are given by $\xi^t \in [n]$ and $\xi^{t-1/2} \in \mathbb{R}$. The state space is $\mathcal{S} = \mathbb{R}_{\ge 0} \times \{\emptyset, \mathtt{a}, \mathtt{r}\}$,

Table 1. Example with Seven Assortments

Parameter	Products	High	High-discount	Low	Low-discount	Only $R2$	Only $R1$	Stock-out
$f_{j\alpha}$	{R1}	5	5	3	3	∞	5	∞
	{R2}	5	5	3	3	5	∞	∞
	{R1,R2}	10	9	6	5	∞	∞	∞
$p_j(\alpha)$	{R1}	0.2	0.2	0.3	0.3	0	0.2	0
	{R2}	0.2	0.2	0.3	0.3	0.2	0	0
	{R1,R2}	0.1	0.15	0.2	0.25	0	0	0

Note. We consider high/low prices with and without bundling discount (i.e., buying $\{R1, R2\}$ is cheaper than buying each individually). The other assortments can be used if items sell out.

where the first component is the remaining knapsack capacity. At a first stage, given a state of the form $s = (b, \emptyset)$, we choose action $\diamond \in \{\mathtt{a}, \mathtt{r}\}$, reducing the capacity if $\diamond = \mathtt{a}$. At the second stage, the state is of the form $s = (b, \diamond)$ with $\diamond \in \{\mathtt{a}, \mathtt{r}\}$, and we collect the reward only if $\diamond = \mathtt{a}$. Formally, the rewards are $\mathcal{R}((b, \mathtt{a}), \xi^{t-1/2}, \emptyset) = \xi^{t-1/2}$ and $\mathcal{R}((b, \mathtt{r}), \xi^{t-1/2}, \emptyset) = 0$.

6.1. Offline Benchmark and Online Policy for Distribution-Agnostic Online Knapsack

To define Offline, φ and $\hat{\varphi}$, consider the following LP parametrized by $(b, \mathbf{y}, \mathbf{z}) \in \mathbb{R}_{\geq 0} \times \mathbb{R}^n \times \mathbb{R}^n_{\geq 0}$:

$$\begin{aligned} (P[b, \mathbf{y}, \mathbf{z}]) \quad \text{maximize:} \quad & \sum_j y_j x_j \\ \text{subject to:} \quad & \sum_j w_j x_{j\mathtt{a}} \leq b, \\ & x_{j\mathtt{a}} + x_{j\mathtt{r}} = z_j \quad \forall j \in [n], \\ & \mathbf{x} \geq 0. \end{aligned} \tag{15}$$

Note that if the average rewards $\mathbf{r}$ were known, then setting $\mathbf{y} = \mathbf{r}$, we get the LP relaxation of Equation (1) for the baseline OnlineKnapsack. Moreover, for any $\mathbf{r}$, the optimal LP solution sorts types by their "bang for the buck" ratios r_j/w_j, and accepts them greedily. In particular, the solution only requires knowing the ranking induced by $\mathbf{r}$.

6.1.1. Offline Benchmark and Relaxed Value Function.

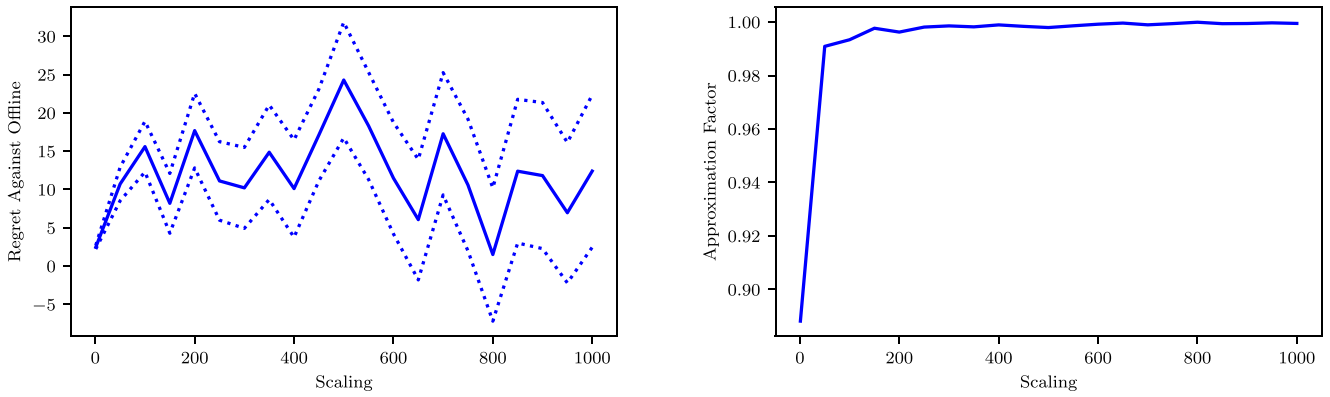
In this setting, we define Offline as the controller that knows the number of arrivals Z_j^T for each j, and also knows the ranking of the types (i.e., knows $r_j/w_j \; \forall j \in [n]$).

Formally, Offline is defined via the filtration $\mathcal{G}_t = \sigma(\{\xi^t : t \in [T]\} \cup \{\xi^\tau : \tau \geq t\})$. This is a canonical filtration (see Definition 1) with variables $(G_\theta : \theta \in \Theta) = (\xi^t : t \in [T])$. Observe that the future rewards, corresponding to times $t - 1/2$, are not revealed. Moreover, the relaxed value is defined as $\varphi(t, s|\mathcal{G}_t) = P[b, \mathbf{r}, Z^t]$ for first stages and

$$\varphi(t - 1/2, s|\mathcal{G}_t) = \begin{cases} P[b, \mathbf{r}, Z^{t-1}] & \diamond = \mathtt{r} \\ \xi^{t-1/2} + P[b, \mathbf{r}, Z^{t-1}] & \diamond = \mathtt{a}. \end{cases} \tag{16}$$

Remark 3 (MDP Relaxations for Distribution-Agnostic Settings)**.** We note here that the underlying problem in this setting does not directly admit an MDP, as the distribution of rewards is unknown. However, once we reveal the arrivals to Offline, the relaxation does admit a well-defined MDP. By benchmarking against Offline, we bypass the need to explicitly formulate

Figure 7. (Color online) Performance of Pricing rabbi with Customer Choice (See Table 1)



Notes. We set the horizon as $T = 10k$ and the inventory as $(R1, R2) = (3k, 2k)$, and vary scaling parameter $k = 1, \ldots, 1{,}000$. (Left) Regret against Offline, with 90% confidence intervals. (Right) Approximation ratio of rabbi against DP.

an Online control problem with distribution learning in this setting.

6.1.2. Value Function Estimate and Online Policy.

Recall we define R_j^t to be the empirical average of the observed rewards for type-j with t periods to go. We define the estimated value as $\hat{\varphi} = P[B^t, R^t, \mathbb{E}[Z^t]]$, resulting in the corresponding online policy given in Algorithm 4.

Algorithm 4 (Learning RABBI)
Input: Access to solutions of $(P[b, \mathbf{y}, \mathbf{z}])$
Output: Sequence of decisions for Online.

1: Set $B^T \leftarrow B$ as the given initial state and R^T as the single sample of each j.
2: **for** $t \in \{T, T-1, \ldots, 1\}$ **do**
3: Compute X^t, an optimal solution to $(P[B^t, R^t, \mathbb{E}[Z^t]])$.
4: Observe the arrival type (context), say $\xi^t = j$, and take any action $\hat{U}^t \in \arg\max_{u=\mathtt{a},\mathtt{r}} \{X_{ju}^t\}$.
5: If $\hat{U}^t = \mathtt{a}$, collect random reward R_j and reduce the budget $B^{t-1} \leftarrow B^t - w_j$. Else, $B^{t-1} \leftarrow B^t$.
6: Update empirical averages R^{t-1} based on R^t and the observation R_j.

6.2. Regret Analysis for Distribution-Agnostic Online Knapsack

As in the earlier sections, we first demonstrate that φ satisfies the Bellman inequalities.

Lemma 8. *The relaxation φ defined in (16) satisfies the Bellman inequalities with the following exclusion sets:*

$$\mathcal{B}(t,b) = \{\omega \in \Omega : \nexists X \text{ solving } (P[b, \mathbf{r}, Z^t]) \text{ s.t. } X_{\xi^t,\mathtt{a}} \geq 1 \text{ or } X_{\xi^t,\mathtt{r}} \geq 1\}.$$

Proof. The initial ordering in Definition 2 follows from an argument identical to that of Lemma 2. The monotonicity property follows from Proposition EC.1 in the online appendix. □

To complete the proof of Theorem 4, we need to characterize the information loss under Algorithm 4. The relaxation relies on the knowledge of $\mathbf{r}$ (the true expectation) and Z^t. The natural estimators are the empirical averages R^t and expectation $\mu(t) = \mathbb{E}[Z^t]$, respectively. Specifically, we use maximizers X^t of $(P[b, R^t, \mu(t)])$ to guess those of $(P[b, \mathbf{r}, Z^t])$.

The overall regret bound is $r_\varphi(\text{Regret}_1 + \text{Regret}_2)$, where Regret_1 and Regret_2 are two specific sources of error. When the estimators R^t of $\mathbf{r}$ are accurate enough, the error is Regret_1 and is attributed to the incorrect guess of a satisfying action, that is, Regret_1 is an algorithmic regret. The second term, Regret_2, is the error that arises from insufficient accuracy of R^t, that is, Regret_2 is the learning regret. The maximum loss satisfies $r_\varphi \leq \max_{j,i}\{w_i r_j / w_j - r_i\}$ and we can show that

$$\text{Regret}_1 \leq 2 \sum_j \frac{(w_{\max}/w_j)^2}{p_j} \quad \text{and}$$

$$\text{Regret}_2 \leq 16 \sum_j \frac{1}{p_j (w_j \delta)^2}.$$

In sum, the regret is bounded by $(\max_{j,i}\{w_i r_j / w_j - r_i\}) \cdot (2\sum_j \frac{(w_{\max}/w_j)^2}{p_j} + 16 \sum_j \frac{1}{p_j (w_j\delta)^2})$.

Remark 4 (Non-i.i.d Arrival Processes)**.** We used the i.i.d. arrival structure to bound two quantities in the proof of Theorem 4: (1) $\mathbb{P}[\|Z^t - \mathbb{E}[Z^t]\| \geq c\mathbb{E}[Z^t]]$, and (2) $\mathbb{E}[e^{-cN_j^t}]$, where, recall, N_j^t is the number of type-j observations. The result holds for other arrival processes that admit these tail bounds.

6.3. Censored Feedback

We consider now the case where only accepted arrivals reveal their reward. We retain the assumption of Theorem 4 that there is a separation $\delta > 0$: $\bar{r}_j - \bar{r}_{j'} \geq \delta$ for all $j \neq j'$, where $\bar{r}_j = \mathbb{E}[R_j]/w_j$.

In the absence of full feedback, we will introduce a unified approach to obtaining the optimal regret (up to constant factors), that takes the learning method as a plug-in. The learning algorithm will decide between explore or exploit actions. Examples of learning algorithms that also give bounds that are explicit in t include modifications of UCB (Wu et al. 2015), ε-greedy or simply to set apart some time for exploration (see Corollary 1).

Recall that $\sigma : [n] \to [n]$ is the ordering of $[n]$ w.r.t. the ratios $\bar{r}_j = r_j/w_j$ and $\hat{\sigma}^t : [n] \to [n]$ is the ordering w.r.t. ratios $\bar{R}_j^t = R_j^t/w_j$. The discrepancy $\mathbb{P}[\sigma \neq \hat{\sigma}^t]$ depends on the plug-in learning algorithm (henceforth Bandits). Bandits receives as inputs the current state S^t (remaining capacity), time, and the natural filtration $\mathcal{F}_t$. The output of Bandits is an action in $\{\texttt{explore}, \texttt{exploit}\}$. If the action is `explore`, we accept the current arrival in order to gather information, otherwise we call our algorithm to decide, as summarized in Algorithm 5. Note that $\mathcal{F}_t$ has information only on the observed rewards, that is, accepted items.

Algorithm 5. Bandits RABBI
Input: Access to Bandits and Algorithm 4.
Output: Sequence of decisions for Online.

1: Set S^T as the given initial state
2: **for** $t = T, \ldots, 1$ **do**
3: Observe input ξ^t and let $U \leftarrow \text{Bandits}(T, t, S^t, \mathcal{F}_t)$.
4: If $U = \texttt{explore}$, accept the arrival.
5: If $U = \texttt{exploit}$, take the action given by Algorithm 4.
6: Update state $S^{t-1} \leftarrow S^t - w_{\xi^t}$ if accept or $S^{t-1} \leftarrow S^t$ if reject.

Theorem 7. *Let* Regret_1 *be the regret of Algorithm 4, as given in Theorem 4. Define the indicators* $explore_t, exploit_t$ *which denote the output of* BANDITS *at time t. The regret of Algorithm 5 is at most* $r_\varphi M$*, where*

$$M = \text{Regret}_1 + \mathbb{E}\left[\sum_t explore_t\right] + \mathbb{E}\left[\sum_t \mathbb{P}[\sigma \neq \hat{\sigma}^t] exploit_t\right].$$

The expected regret of Algorithm 5 is thus bounded by the regret of Algorithm 4 in the full feedback setting, plus a quantity controlled by BANDITS. In the periods where BANDITS says explore (which, in particular, implies accepting the item), the decision might be the wrong one (i.e., different from OFFLINE's). We upper bound this by the number of exploration periods. This is the second term in M. The decision might also be wrong if BANDITS says exploit (in which case we call Algorithm 4), but the (learned) ranking at time t, $\hat{\sigma}^t$, is different from σ^t. This is the last term in M. Finally, even if the learned ranking is correct, exploit can lead to the wrong guess by Algorithm 4 because the arrival process is uncertain. This is the first term in M.

Corollary 1 uses a naive BANDITS, which explores until obtaining $\Omega(\log T)$ samples and achieves the optimal (i.e., logarithmic) regret scaling. The constants may be improved by changing the BANDITS module we use. Any such algorithm has the guarantee given by Theorem 7. With the naive BANDITS, the bound follows from a generalization of coupon collector (Shank and Yang 2013).

Corollary 1. *If we first obtain* $\frac{8}{(w_j\delta)^2}\log T$ *samples of every type j, then we can obtain* $O(\log T)$ *regret, which is optimal up to constant factors.*

7. Concluding Remarks

We developed a framework that provides rigorous support to the use of simple optimization problems as a basis for online resolving algorithms. The framework is based on using a carefully chosen offline benchmark, which guides the online algorithm. The regret bounds then follow from our use of Bellman inequalities and a useful distinction between Bellman loss and information loss.

As is often the case in approximate dynamic programming, the identification of a function φ satisfying the Bellman inequalities requires some ad hoc creativity but, as our examples illustrate, is often rather intuitive. In online Appendix EC.1, we provide sufficient conditions, applicable to cases where φ has a natural linear representation, to verify the Bellman inequalities. These conditions are intuitive and likely to hold for a variety of resource-allocation problems. Importantly, once such a function is identified, our RABBI framework provides a way of obtaining online policies from φ, and corresponding regret bounds.

We illustrate our framework on three settings. First, we consider online probing, which serves as an instance of a larger family of two-stage decision problems, wherein there is an inherent trade-off between getting refined information, and the cost of obtaining it. Next, we consider dynamic pricing, which is a well-studied problem, and is representative of settings where rewards and transitions are random. Finally, our study of online contextual bandits with knapsacks showcases a separation of the underlying combinatorial problem from the parameter estimation problem.

It is our hope that this structured framework will be useful in developing online algorithms for other problems, whether these are extensions of those we studied here or completely different.

References

Agrawal S, Devanur N (2016) Linear contextual bandits with knapsacks. *Proc. 30th Internat. Conf. Neural Inform. Processing Systems* (Curran Associates Inc, Red Hook, NY), 3450–3458.

Alaei S (2014) Bayesian combinatorial auctions: Expanding single buyer mechanisms to many buyers. *SIAM J. Comput.* 43(2):930–972.

Arlotto A, Gurvich I (2019) Uniformly bounded regret in the multisecretary problem. *Stochastic Systems* 9(3):231–260.

Babaioff M, Dughmi S, Kleinberg R, Slivkins A (2015) Dynamic pricing with limited supply. *ACM Trans. Econom. Comput.* 3(1): article 4, https://doi.org/10.1145/2559152.

Badanidiyuru A, Kleinberg R, Slivkins A (2018) Bandits with knapsacks. *J. ACM* 65(3):Article 13.

Badanidiyuru A, Langford J, Slivkins A (2014) Resourceful contextual bandits. *Conf. Learn. Theory*, 1109–1134.

Balseiro SR, Brown DB (2019) Approximations to stochastic dynamic programs via information relaxation duality. *Oper. Res.* 67(2): 577–597.

Banerjee S, Freund D (2020) Uniform loss algorithms for online stochastic decision-making with applications to bin packing. *Abstr. 2020 SIGMETRICS/Performance Joint Internat. Conf. Measurement Model. Comput. Systems* (Association for Computing Machinery, New York), 1–2.

Boucheron S, Lugosi G, Massart P (2013) *Concentration Inequalities: A Nonasymptotic Theory of Independence* (Oxford University Press, Oxford, UK).

Brown D, Smith J, Sun P (2010) Information relaxations and duality in stochastic dynamic programs. *Oper. Res.* 58(4):785–801.

Bubeck S, Cesa-Bianchi N (2012) Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations Trends Machine Learn.* 5(1):1–122.

Bubeck S, Perchet V, Rigollet P (2013) Bounded regret in stochastic multi-armed bandits. *Proc. Conf. Learn. Theory*, PLMR 30: 122–134.

Buchbinder N, Jain K, Singh M (2014) Secretary problems via linear programming. *Math. Oper. Res.* 39(1):190–206.

Bumpensanti P, Wang H (2020) A re-solving heuristic for dynamic resource allocation with uniformly bounded revenue loss. *Management Sci.*, ePub ahead of print March 16, https://doi.org/10.1287/mnsc.2019.3365.

Chen Q, Jasin S, Duenyas I (2019) Nonparametric self-adjusting control for joint learning and optimization of multiproduct pricing with finite resource capacity. *Math. Oper. Res.* 44(2): 377–766.

Chugg B, Maehara T (2019) Submodular stochastic probing with prices. *Proc. 66th Internat. Conf. Control, Decision Inform. Tech.* (IEEE, Paris), 60–66.

Correa J, Foncea P, Hoeksma R, Oosterwijk T, Vredeveld T (2017) Posted price mechanisms for a random stream of customers. *Proc. ACM Conf. Econom. Comput.*, 169–186.

Düetting P, Feldman M, Kesselheim T, Lucier B (2017) Prophet inequalities made easy: Stochastic optimization by pricing non-stochastic inputs. *Proc. 58th Annual Symp. Foundations Comput. Sci.* (IEEE, Berkeley, CA), 540–551.

Gallego G, Van Ryzin G (1997) A multiproduct dynamic pricing problem and its applications to network yield management. *Oper. Res.* 45(1):24–41.

Gupta A, Nagarajan V (2013) A stochastic probing problem with applications. *Proc. Internat. Conf. Integer Programming Combin. Optim.* (Springer, Berlin, Heidelberg), 205–216.

Gupta A, Nagarajan V, Singla S (2016) Algorithms and adaptivity gaps for stochastic probing. *Proc. 27th Annual ACM-SIAM Symp. Discrete Algorithms*, 1731–1747.

Jasin S (2014) Reoptimization and self-adjusting price control for network revenue management. *Oper. Res.* 62(5):1168–1178.

Jasin S, Kumar S (2012) A re-solving heuristic with bounded revenue loss for network revenue management with customer choice. *Math. Oper. Res.* 37(2):313–345.

Kleinberg R, Weinberg SM (2012) Matroid prophet inequalities. *Proc. 44th Annual ACM Symp. Theory Comput.*, 123–136.

Mangasarian O, Shiau T (1987) Lipschitz continuity of solutions of linear inequalities, programs and complementarity problems. *SIAM J. Control Optim.* 25(3):583–595.

Massart P (1990) The tight constant in the Dvoretzky-Kiefer-Wolfowitz inequality. *Ann. Probab.* 18(3):1269–1283.

Shank N, Yang H (2013) Coupon collector problem for non-uniform coupons and random quotas. *Electron. J. Combin.* 20(2), doi:.

Singla S (2018) *The price of information in combinatorial optimization. Proc. 29th Annual ACM-SIAM Symp. Discrete Algorithms*, 2523–2532.

Talluri KT, Van Ryzin GJ (2006) *The Theory and Practice of Revenue Management*, vol. 68 (Springer Science & Business Media, New York).

Vera A, Banerjee S (2020) The Bayesian prophet: A low-regret framework for online decision making. *Management Sci.*, ePub ahead of print October 5, https://doi.org/10.1287/mnsc.2020.3624.

Weitzman ML (1979) Optimal search for the best alternative. *Econometrica* 47(3):641–654.

Wu H, Srikant R, Liu X, Jiang C (2015) Algorithms with logarithmic or sublinear regret for constrained contextual bandits. *Adv. Neural Inform. Processing Systems* 28:433–441.

Alberto Vera received his PhD in Operations Research from Cornell in 2020. He is currently a research scientist at Amazon working on resource allocation problems.

Sid Banerjee is an assistant professor in the School of Operations Research at Cornell, working on topics at the intersection of data-driven decision-making, network algorithms and market design. He received his PhD in 2013 from the ECE Department at UT Austin, and was a postdoctoral researcher in the Social Algorithms Lab at Stanford from 2013-2015

Itai Gurvich is a professor at Northwestern's Kellogg School of Management. His research focuses on the performance analysis and optimization of processing networks.