Constant Regret in Online Allocation: On the Sufficiency of a Single Historical Trace

Siddhartha Banerjee [1] **Itai Gurvich** [2] **Alberto Vera** [2]

Abstract

We consider online decision-making problems where resources are allocated dynamically to a stochastic stream of requests, and decisions are made to maximize reward while satisfying a set of constraints. We propose and analyze a simple algorithm that uses only historical data, i.e., traces (sample paths) of the stochastic process. We prove that, in a large family of problems, which includes as special cases online packing and online matching, the algorithm has near-optimal performance with the minimum possible sample-complexity; in particular, it obtains *constant regret with as few as one trace*. The algorithm is agnostic of the generative model of arrivals; however, the results hold even under time-varying and correlated arrival processes. Finally, even in settings beyond our theoretical guarantees, our framework generates data-friendly algorithms that match and beat the performance of specialized state-of-the-art algorithms in simulations.

1. Introduction

Online decision-making problems arise in many domains, and it is fundamental to understand the role of historical data in such problems, and how to use it for designing algorithms. In this work, we study this question in the context of online resource allocation problems. We focus on a general setting where a controller faces a stream of requests of various types over a finite horizon T, and must decide dynamically how to allocate resources, while collecting a reward associated with the request's type. There is a finite initial stock (inventory) of resources, which is depleted as requests are accepted. Many well-studied problems fit in this description (see Section 2.2 for details).

[1]Department of Operations Research, Cornell University, Ithaca, NY [2]Department of Operations Research, Cornell University, New York, NY. The work of the second and third authors was supported by DoD grant W911NF-20-C-0008. Correspondence to: S Banerjee, I Gurvich, A Vera <{sbanerjee,gurvich,aav39}@cornell.edu>.

The main source of uncertainty in online resource-allocation is in the request-arrival process, and different approaches to online decision-making make different assumptions about how this uncertainty is realized. Moreover, these problems are typically not just run once, but repeatedly over many length-T *episodes*, and at any time, a controller typically has data of arrivals from past episodes, as well as of past arrivals in the current episode. To distinguish between these, we henceforth refer to data from previous episodes as *historical traces*, and data from the current episode as the *online trace*.

In practical settings, arrival processes may vary across different episodes (due to weekly/seasonal variations, etc.). What is often true, however, is that the controller has *some historical traces which are representative of the current episode*. For example, a network router can use request traces over the last hour; smart-grid operators can use electricity demand from the same hour yesterday; a rideshare platform can use ride requests from the same day in the previous week; hotels and airlines can use reservations from last year's holiday period to model this year's holidays. One can model this by assuming arrivals in an episode are drawn from some unknown underlying process, and *the controller is given a single (or few) historical trace drawn from the same process*. How can we use this to design good policies?

Existing methods for online decision-making take two broad approaches for dealing with uncertainty: (i) Bayesian approaches posit an underlying stochastic model for the uncertainty, and (ii) worst-case approaches view decision-making as a game against an adversary generating the uncertainty. In terms of data dependence, these approaches lie at two ends of a spectrum. Bayesian approaches like approximate dynamic programming (ADP) and reinforcement learning (RL) require extensive historical traces to learn good policies; worst-case approaches like online learning and competitive analysis only use the online trace to guide decisions. A natural question is if there are variants of these approaches that can leverage the problem structure to find good policies using only a few historical traces.

In this paper, we propose a novel policy that is able to leverage few traces to get strong performance guarantees in theory and practice. Our algorithm builds on a recent framework called the Bellman inequalities (Vera et al., 2019),

which uses a sequence of offline optimization programs to get constant regret algorithms for online resource allocation. Surprisingly, we show that we only need *a single trace of historical data to obtain constant regret* (i.e., independent of the size of the state-space and length of horizon T) in many problem settings, and under a wide class of processes including non-stationary and correlated processes.

In adapting the algorithm from (Vera et al., 2019) to work with traces, we bring out a natural robustness property of this paradigm relative to other ADP approaches. Informally, the core idea behind our proposed algorithm is that, by resolving offline relaxations, a controller can aggregate uncertainty in a way that reduces sensitivity to estimation errors. To highlight this, we show that we outperform common ADP approaches. In particular, standard approaches to data-driven ADP are based on first estimating the parameters of the arrival model (or more directly, the Q-functions for each state (Mannor et al., 2007)) and then running an ADP algorithm with the estimated model as input. A hurdle in this approach is how to handle ties, i.e., when multiple actions look equally good relative to the (estimated) value function. We show that the tie-breaking rule is crucial for good performance, and standard approaches, in particular randomized rules, lead to sub-optimal performance (cf. Fig. 1). Our algorithm provides one such family of data-driven tie-breaking rules that ensures good performance.

Finally, the intuition behind our theoretical bounds also informs the design of algorithms for problems that lie beyond the theory. We illustrate this in the context of reusable resources (e.g. inventory control, allocation of cloud resources and hotel rooms, etc.) where it is known that no online algorithm can have constant regret. Our framework generates a straightforward algorithm that is computationally efficient and showcases strong performance, beating other state-of-the-art specialized algorithms.

1.1. Related Work

As we mention above, our work is closely related to ADP approaches for online allocation problems (Gallego & Van Ryzin, 1994; Jasin & Kumar, 2012; Arlotto & Gurvich, 2019; Vera & Banerjee, 2019; Bumpensanti & Wang, 2018), bandit approaches with iid arrivals from an unknown distribution (Badanidiyuru et al., 2013), and worst-case models for online packing (Buchbinder et al., 2009; Devanur et al., 2019), and more generally, online convex optimization (Agrawal & Devanur, 2014; Agrawal et al., 2014). These methods do not naturally use traces, and hence guarantees are difficult to compare; nevertheless we numerically demonstrate that our algorithm performs much better than (i) state-of-the-art parametric ADP approaches with full model information, and (ii) worst-case models even under complex input processes.

Our work is closely related to the literature on *prophet inequalities* (Kleinberg & Weinberg, 2012) and in particular, sample-based variants (Azar et al., 2014; Rubinstein et al., 2020). These works however focus on worst-case distributions, and have poor performance for typical distributions. Our work builds on a more recent approach towards getting distribution-dependent prophet inequalities, and in particular, on the work of (Arlotto & Gurvich, 2019) for multi-secretary problems, and Vera & Banerjee (2019); Vera et al. (2019) for general online allocation problems. The general algorithm we introduce here adapts the RABBI algorithm in (Vera et al., 2019) to incorporate historical traces.

2. Problem Setting and Summary of Results

We consider the following general online resource-allocation problem, which takes place over a finite horizon of T periods. We use T to denote the first period, and index time in terms of *periods to go*, i.e., $t \in \{T, T-1, \ldots, 1\}$.

We are given a set of d resource types, and an initial state $S^T \in \mathbb{N}^d$ comprising of an initial stock of S_i^T for each resource type $i \in [d]$. In each period, an incoming request $\xi^t \in [n]$ is generated from a set of n request types via an *exogenous stochastic process*. Faced with a request of type j, the controller can choose one from a set $\mathcal{U}$ of controls, with associated rewards r_{uj} for $u \in \mathcal{U}$ and $j \in [n]$. A control $u \in \mathcal{U}$ applied to a request type j depletes resources $A_{uj} \in \mathbb{Z}^d$, i.e., if the current stock is S, then after applying u the levels change to $S - A_{uj}$.

The above problem setting encapsulates many important Markov decision processes (MDPs) (cf. Section 2.2). A critical aspect of all settings we study is that the randomness arises via an exogenous input process $(\xi^t : t \in [T])$, which is independent of the state and actions of the controller. Note that If the controller has full knowledge of the underlying generative process, then it can use this to find optimal control policies via ADP techniques.

Our work instead assumes that the controller's information about the process is only *a limited number of historical traces*. Formally, we assume the controller is provided with K historical traces $\mathcal{H} = \{(\widehat{\xi}^{t,k} : t \in [T]) : k \in [K]\}$ of the exogenous request process, which are sampled from the underlying generative model. We focus on the case of $K = 1$, i.e., where the controller has only a single historical trace. Using this, the controller makes real-time decisions on new incoming requests. Henceforth we consistently refer to $(\xi^t : t \in [T])$ as the *online trace* and $(\widehat{\xi}^{t,k} : t \in [T])$ as *historical traces*.

Unlike in MDPs, in the context of online optimization with historical traces there is no clear notion of an optimal policy. Nevertheless, a natural benchmark for any policy is the so-called *prophet* (or offline/hindsight) benchmark – the

performance of a controller that has full information of the online trace. Obviously, the offline controller has no use for the historical traces as it sees the full online trace, and uses it to solve the following problem:

$$\begin{array}{ll} \max_{\mathbf{x}\in\mathbb{R}^{[n]\times[T]\times\mathcal{U}}_{\geq 0}} & h(\mathbf{x};\xi^T,\ldots,\xi^1) \\ \text{s.t.} & g(\mathbf{x};S^T,\xi^T,\ldots,\xi^1)\geq \mathbf{0}. \end{array} \tag{1}$$

In most problem of interest, the decision variable $x^t_{u,\xi}$ is binary and represents if the control u is used at time t when an input of type ξ is presented. We assume that the functions h and g, encoding the structure of the problem, are known. The following example, also referred to as unit-weight knapsack and online k-uniform matroid, suffices to understand our main results (see Section 2.2 for additional problems).

Example 2.1 (Multi-Secretary). Job candidates arrive sequentially for the selection to one of $S^T \in \mathbb{N}$ vacant positions. The candidate arriving at time t has ability (reward) r_j with probability p^t_j, independent of other candidates. The controller must irrevocably accept ($\mathtt{a}$) or reject ($\mathtt{r}$) the candidate. The goal is to select S^T candidates so as to maximize the total reward.

In this setting, the offline problem can be described by the total number of accepted and rejected type-j arrivals, $y_{\mathtt{a},j}$ and $y_{\mathtt{r},j}$ respectively. Let Z^t_j denote the total number of type-j arrivals over the last t periods, and S^t denote the number of vacant slots at the start of period t. Now Eq. (1) (computed at time-to-go t) takes the following form:

$$\max_{\mathbf{y}\geq 0}\left\{\sum_{j\in[n]} r_j y_{\mathtt{a},j} : \sum_{j\in[n]} y_{\mathtt{a},j}\leq S^t, y_{\mathtt{a},j}+y_{\mathtt{r},j}=Z^t_j\,\forall\, j\right\}. \tag{2}$$

Our approach is to solve Eq. (2) with $\widehat{\xi}^{t:1}$ as input, i.e., obtain cumulative arrivals $\widehat{Z}^t$ from the historical traces, and use the solution $\{\widehat{y}_{a,j},\widehat{y}_{r,j}\}$ as *scores* given to the actions accept and reject respectively. We take the action with the highest score in each period. This idea, which forms the basis of our proposed RABBI algorithm, leads to significant improvements in the regret, as we quantify both in terms of theoretical guarantees and simulations.

2.1. Overview of our Results

We present the performance guarantees of our algorithm, formally defined in Section 3, which is a data-integrated adaption of RABBI introduced in (Vera et al., 2019). We provide here high-level statements of the main results, their formal statements and analysis is in Section 4.

For a given online trace $\xi^{T:1}$, let $V^{\text{off}}(S^T,\xi^{T:1})$ denote the reward collected by the offline controller, i.e., the objective value of the optimization problem in Eq. (1). An algorithm ALG that uses the historical traces in $\mathcal{H}=(\widehat{\xi}^{T:1,k}: k\in[K])$, collects total reward $V^{\text{ALG}}(T,S^T,\xi^{T:1},\mathcal{H})$, and incurs a *regret* defined as

$$\text{Reg}^{\text{ALG}}(T,S^T)=\mathbb{E}\left[V^{\text{off}}(S^T,\xi^{T:1})-V^{\text{ALG}}(T,S^T,\xi^{T:1},\mathcal{H})\right],$$

where the expectation is taken w.r.t. the historical and online traces as well as ALG's internal randomness (if any).

Our main algorithmic contribution is a general data-driven policy for online resource allocation problems, which we refer to as the RABBI algorithm (presented in Section 3). The basic idea behind RABBI is to act greedily based on action-scores computed via Eq. (1) using the historical trace (see discussion in previous section on the multi-secretary problem). The following theorem encapsulates our main performance guarantee for this algorithm.

Theorem 2.2. *For a large class of online allocation problems (cf. Section 2.2), and stochastic arrival processes (cf. Section 2.3),* RABBI *achieves constant regret with $K=1$ historical trace. In other words, $\text{Reg}^{\text{RABBI}}(T,S^T)\leq\rho$ for some constant ρ independent of T and S^T.*

Notice that constant regret implies, in particular, an approximation ratio of $1-\mathcal{O}(\frac{1}{V^{\text{off}}})$, where $V^{\text{off}}=\mathbb{E}[V^{\text{off}}(S^T,\xi^{T:1})]$; see for example Figure 1 in Section 5.

The importance of score-based decisions is highlighted when we consider the natural way to incorporate historical data into a Dynamic Program (DP). A standard approach is to estimate the parameters and subsequently run the DP. Formally, consider an independent time-varying arrival process: $\mathbb{P}[\xi^t=j]=p^t_j$, with $\mathbf{p}\in\mathbb{R}^{n\times T}_{\geq 0}$ unknown parameters. Let $V^t(S,j;\mathbf{p})$ represent the total value if there are t periods to go, the current input is $\xi^t=j$, the stock levels are S, and the probabilities are $\mathbf{p}$, then the value function is updated according to:

$$V^t(S,j;\mathbf{p})=\max_{u\in\mathcal{U}}\left\{r_{uj}+\sum_{j'\in[n]} p^{t-1}_{j'}V^{t-1}(S-A_{uj},j';\mathbf{p})\right\}.$$

With historical traces, the parameters $\mathbf{p}$ are estimated from historical data, hence with K traces DP uses the empirical averages $\widehat{p}^t_j=\frac{|\{k\in K:\widehat{\xi}^{t,k}=j\}|}{K}$.

When executing the policy that arises from the DP, there is a degree of freedom in determining how to break ties — i.e. how to choose between controls that are equally optimal relative to the *estimated* value function. That is, if there is a unique control that is optimal relative to $V^t(S,j;\widehat{\mathbf{p}})$, then that control must be taken. Conversely, if there are multiple such controls, one must choose between them.

Viewed in the above framework, RABBI can be used as follows: when DP encounters a tie, it calls a single resolve of RABBI and uses its scores to choose among actions. We prove that DP, equipped with this tie-breaking rule, achieves constant regret for all the problems for which

RABBI achieves constant regret. In contrast, other natural tie-breaking rules, in particular randomized tie-breaking, may have $\omega(1)$ regret, see Fig. 1.

Theorem 2.3. *For a large family of problems and independent time-varying processes,* DP *with the tie-breaking rule from* RABBI *achieves constant regret with $K = 1$ historical trace. In other words, there is a constant ρ independent of T and S^T such that $\mathrm{Reg}^{\mathrm{DP}}(T, S^T) \leq \rho$.*

2.2. Application to Specific Problems

We now discuss several problems of practical interest that can be encoded via our general setting. The controls $\mathcal{U}$ correspond to allocation of resources and the specific dynamics give rise to different problems, whereas the following description is common to all problems. Let us denote Z_j^t as the cumulative number of type-j requests in the last t periods, i.e., $Z_j^t \coloneqq \sum_{\tau=1}^{t} \mathbb{1}_{\{\xi^\tau = j\}}$. Additionally, if x_{uj}^t represents that the control u was used at t when an input type j was presented, then all the problems we consider satisfy the following natural conditions, which are interpreted as "some control is used at every time period":

$$\sum_{u \in \mathcal{U}} \sum_{\tau=1}^{t} x_{uj}^{\tau} = Z_j^t \quad \forall j \in [n], t \in [T]. \tag{3}$$

Settings Admitting Uniform Regret Guarantees The following represent important special cases of settings in which our main theorem, Theorem 2.2, guarantees that RABBI incurs $O(1)$ regret (in particular, they satisfy the requirements R1-R4 given in in Section 4).

1. Online Packing (Network Revenue Management): Each type $j \in [n]$ is associated with a vector $A_j \in \{0,1\}^d$ of resources and a reward $r_j \geq 0$. At each time, we must decide if we allocate all the resources requested (encoded in A_j) or if we reject the request. Hence, the control set has two actions, $\mathcal{U} = \{\mathtt{a}, \mathtt{r}\}$. The function g encodes the following constraint for all resources $i \in [d]$: $\sum_j A_{ij} \sum_{t=1}^{T} x_j^t \leq S_i^T$, meaning that the amount of allocated resources do not exceed the initial stock.
2. Online Matching: Each type $j \in [n]$ has associated a reward vector $r_j \in \mathbb{R}^d_{\geq 0}$. At each time, we must decide to which resource i we match the current arrival j, which would generate reward r_{ij}. The control set is $\mathcal{U} = [d]$. If $r_{ij} = 0$, we interpret that j cannot be assigned a resource type i. The function g encodes $\sum_j \sum_{t=1}^{T} x_{ij}^t \leq S_i^T$.
3. Generalized Assignment: each $i \in [d]$ represents a bin and, if we place a type j in bin i, it uses $a_{ij} \geq 0$ space while generating a reward of r_{ij}. The control set is again $\mathcal{U} = [d]$ and g encodes $\sum_j \sum_{t=1}^{T} a_{ij} x_{ij}^t \leq S_i^T$.
4. Online Probing: Packing problems with unknown random rewards, where the controller can probe up to S_0^T requests to reveal their true reward before deciding on a control. The available controls are 'accept', 'probe', and 'reject'. The functions h and g are again linear objectives and constraints (cf. (Vera et al., 2019) for details).

Settings we Study Numerically A common feature of the above settings is that actions are *exchangeable* across arrivals of the same type – the objective and constraints only depend on the number of times an action u is taken for arrival type j, and not when they were taken (this is encoded in requirement R3 in Section 4). In settings where the timing of actions matters, uniform regret guarantees may not be possible. Nevertheless, these settings are still captured by our formulation, and we show via simulations that RABBI has a superior performance even compared to state-of-the-art algorithms crafted specially for these problems. In particular, we consider the following *reusable resource* problems, where allocated resources are returned to the platform after some time.

1. Reusable Matching: As in online matching, type-j arrivals can be matched to at most one resource i for reward r_{ij}, but now the allocated resource is returned after l_j periods. For known distributions, recent work (Rusmevichientong et al., 2020) presents a 0.5-approximation for this setting using ADP with linear basis functions.
2. Reusable Packing: Multidimensional packing problems, where resources are released after some time. To the best of our knowledge, there are no algorithms with provable guarantees for this setting, and it is unclear how to extend other algorithms for this case.

2.3. Stochastic Generative Models for Arrivals

We consider Markov modulated arrival processes, which capture independent time-varying processes as well as other correlated models. Let (M^t) be a Markov chain with state space V and initial distribution $\nu : V \to [0,1]$. The arrival process is modulated by (M^t) as follows: at time t there is an arrival of type j with probability $p_j(M^t)$, where $p_j : V \to [0,1]$, i.e., $\mathbb{P}[\xi^t = j | \mathcal{F}_t] = p_j(M^t)$. In summary, to completely specify the arrival process, we need the law of (M^t) as well as the functions $p_j(\cdot)$, both of which are assumed to be unknown. We also consider case where the Markov chain is hidden.

This covers the case of i.i.d arrivals; the modulation process has just one state ($|V = 1|$), i.e., $p_j(M^t) = p_j$. Additionally, it covers the case of time-varying independent arrivals if we define the chain's state-space be $V = [T]$ and endow it with deterministic transitions $\mathbb{P}[M^{t-1} = t - 1 | M^t] = 1$.

3. The Data-Driven RABBI Algorithm

Recall the offline benchmark problem in Eq. (1). RABBI stipulates resolving, at each time t, a problem with estimated functions $\bar{h}^t \approx h, \bar{g}^t \approx g$ and updated state S^t. Formally, define $\widehat{\varphi}(t, s)$ as the following program:

$$\widehat{\varphi}(t,s) = \begin{array}{ll} \max\limits_{\mathbf{x}\in\mathbb{R}^{[n]\times[T]\times\mathcal{U}}} & \bar{h}^t(\mathbf{x};\xi^T,\ldots,\xi^1) \\ \text{s.t.} & \bar{g}^t(\mathbf{x};s,\xi^T,\ldots,\xi^1)\geq \mathbf{0}, \end{array} \quad (4)$$

where $\bar{h}^t, \bar{g}^t$ are the empirical estimates given by $\bar{h}^t = \frac{\sum_{k\in[K]} h(\mathbf{x};\widehat{\xi}^{t:1,k})}{K}$ and $\bar{g}^t = \frac{\sum_{k\in[K]} g(\mathbf{x};S^t,\widehat{\xi}^{t:1,k})}{K}$.

Let $\widehat{\mathbf{x}}$ denote the optimal solution to Eq. (4), and define *score* $\widehat{y}^t_{u,j} = \sum_{\tau=1}^t \widehat{x}^\tau_{u,j}$ for each action u when faced with a type-j request. The main idea in RABBI is to re-compute these scores in each period, and act greedily with respect to them. The formal algorithm is as follows:

RABBI (Re-solve and Act Based on Bellman Inequalities)

Input: Access to historical traces $(\widehat{\xi}^{t,k} : t\in[T]), k\in[K]$.
Output: Sequence of decisions $\widehat{U}^t$ for each $t\in[T]$.

1: Set S^T as the given initial state
2: **for** $t = T,\ldots,1$ **do**
3: Re-solve $\widehat{\varphi}(t,S^t)$ as in Eq. (4)
Compute scores $\widehat{\mathbf{y}} = \{\widehat{y}^t_{u,j}\}_{u\in\mathcal{U},j\in[n]}$
4: Given ξ^t, choose action $\widehat{U}^t$ with highest score $\widehat{y}^t_{u,\xi^t}$
5: Collect reward $r_{\widehat{U}^t,\xi^t}$. Update $S^{t-1}\leftarrow S^t - A_{\widehat{U}^t,\xi^t}$
6: **end for**

4. Analysis

We now present formal regret guarantees for RABBI based on a single trace. Our results hold for any problem satisfying three properties that encapsulate (i) the structure of the optimization problem, (ii) the predictability of the stochastic process, and (iii) the interaction between the optimization problem and the stochastic process.

Our first requirement is that the offline problem is linear and 'well-posed', in that it captures underlying transitions.

R1 *Well-Posed Problem*: The reward h has the form $h(\mathbf{x};\xi^{t:1}) = \sum_{j,u} x_{ju} r_{ju}$. The function g is linear, and includes the constraints $\sum_u x_{ju} = Z^t_j \,\forall\, j\in[n]$, i.e, an action must be selected for every arrival. For all $u\in\mathcal{U}, j\in[n]$, if control u is applied to type j, then the constraint g captures the resulting depletion of resources; formally, $g(\mathbf{e}^t_{u,j};s,\xi^{t:1}) \leq g(\mathbf{0};s-A_{uj},\xi^{t:1})$.

Next, we require that the stochastic arrival process satisfies a concentration bound, which is a variant of standard bounds based on 'self-normalized exchangeable pairs'.

R2 *All Time Concentration of Arrivals*: $\exists\,\theta_1,\theta_2>0$, s.t.

$$\mathbb{P}[||Z^t-\widehat{Z}^t||_\infty \geq \widehat{Z}^t_k/c] \leq \theta_1 e^{-\frac{\theta_2 t}{c^2}}, \quad \forall k\in[n]. \quad (5)$$

Finally our third condition requires the constraints can be expressed as an additive function of the actions and arrivals.

R3 *Noise Interaction*: The constraint function satisfies $g(\mathbf{x};s,\xi^{t:1}) = g_1(\mathbf{x};s)+g_2(s;\xi^{t:1})$ and $||g_2(s;\xi^{t:1})-g_2(s;\widehat{\xi}^{t:1})||_\infty \leq ||Z^t-\widehat{Z}^t||_\infty$. In other words, the noise interacts additively with the constraints and it is dominated by the demand.

Example 4.1. We show that the multi-secretary problem (Example 2.1) satisfies requirements R1 and R3. Recall that the problem has two sets of constraints, $\sum_j x_{\mathrm{a},j}\leq S^t$ (acceptances do not exceed the available positions) and $x_{\mathrm{a},j}\leq Z^t_j$ (acceptance of j does not exceed arrivals of j). R1 follows by inspection. R2 follows because the constraint that involves noise is $\mathbf{x}_{\mathrm{a}}\leq Z^t$, which is additive in the noise with $g_2(s;\xi^{t:1}) = Z^t$.

In what follows we use the constant κ that identifies the Lipshitz continuity of the underlying LP. Let the matrix $\bar{A}$ be the constraint matrix encoded by g. Then, we write $\kappa = \kappa(\bar{A})$ to be the constant such that for any two right-hand side values $\mathbf{b}^1,\mathbf{b}^2$, the set of optimal solutions $\mathcal{X}(v) := \mathrm{argmax}_{\mathbf{x}}\{\mathbf{r}'\mathbf{x} : \bar{A}\mathbf{x}\leq\mathbf{b}\}$ satisfies $||\mathcal{X}(\mathbf{b}^1)-\mathcal{X}(\mathbf{b}^2)||_\infty \leq \kappa||\mathbf{b}^1-\mathbf{b}^2||_\infty$. The existence of this constant κ follows, e.g., from (Mangasarian & Shiau, 1987).

Theorem 4.2. *Suppose that the optimization problem (1) satisfies the three requirements listed above. Then, the regret of* RABBI *with a single trace is at most* $dr_{\max}|\mathcal{U}|^2\theta_1\kappa^2/\theta_2$, *where* $r_{\max} = \max_{j\in[n],u\in\mathcal{U}} r_{uj}$ *is the maximum reward and* d *is the number of resource types.*

We complement our result by showing that a large family of stochastic processes satisfy Eq. (5).

Proposition 4.3. *For time-varying independent processes with* $p_{jt}\geq\beta$ *for all* j,t, *Eq. (5) is satisfied with* $\theta_1 = 4n$ *and* $\theta_2 = \beta^2/12$, *hence the regret is at most* $\mathcal{O}(dr_{\max}n\kappa^2/\beta^2)$.

The next example emphasises that, if the arrival process is not independent, in general a single trace is insufficient. Motivated by this, we study in Propositions 4.5 and 4.7 particular forms of correlation.

Example 4.4. Consider the arrival process where, at time $t=T$, i.e., at the beginning of the horizon, a fair coin is flipped. With probability $1/2$, we take ξ^t to be independent random variables with $\mathbb{P}[\xi^t=j] = p^t_{j,\text{head}}, j\in[n], t\in[T]$. With the remaining probability $1/2$, $\mathbb{P}[\xi^t=j] = p^t_{j,\text{tail}}$. In this case, with probability $1/2$ the online trace comes from a different distribution that the single historical trace, hence the latter contains no relevant information.

In the following two results, we assume that the underlying Markov chain has a stationary distribution $\pi : V\to[0,1]$. We denote M^t the online chain and $\widehat{M}^t$ the historical chain, modulating the online and historical trace, respectively. We give *parametric results* in terms of (i) the absolute spectral gap and (ii) mixing time of M^t.

For a function $h : V\to\mathbb{R}$ we define $||h||^2_\pi = \int h(v)^2\pi(dv)$ Given a Markov operator P, the absolute spectral gap

is $1-\lambda(P)$, where $\lambda(P) = \sup\{||Ph||_\pi : ||h||_\pi = 1, \int h(v)\pi(dv) = 0\}$; see (Fan et al., 2018).

Proposition 4.5. *Suppose that the arrival process is Markov modulated with a chain M^t that has invariant measure π and absolute spectral gap $1-\lambda$. Then, if M^T and $\widehat{M}^T$ are initialized with π, Eq. (5) holds with $\theta_1 = 4n$ and $\theta_2 = 48\frac{1-\lambda}{1+\lambda}\beta^2$, hence the regret is at most $\mathcal{O}(\frac{dr_{\max}n\kappa^2}{(1-\lambda)\beta^2})$. Furthermore, the chain can be hidden from the controller.*

Example 4.6. The 2-state Markov chain with states $V = \{1,2\}$ and transition matrix $P = \left(\begin{smallmatrix} 1-\delta & \delta \\ \delta & 1-\delta \end{smallmatrix}\right)$ has spectral gap $1-\lambda = 3\delta(1-\delta)$. The modulated arrival process has parameters $p_j(1), p_j(2)$. With a single historical trace, if $\delta \approx 0$, it could be that the online trace evolves with parameters $p_j(1)$ for most periods, whereas the historical trace with $p_j(2)$ for most periods. This captures the intuitive notion that a slower-mixing chain releases less information in a single trace. In other words, if $\delta \approx 0$ a single trace is uninformative. This is reflected in our regret bound: it grows as $1/\delta$ as $\delta \downarrow 0$.

It is, in fact, not necessary for the chain to be initialized with its stationary distribution. With other conditions, it suffices that M^T and $\widehat{M}^T$ are initilaized in the same state. To state this, the mixing time of a Markov chain is given by $t_{\text{mix}} := \min\{t : \sup_{v\in V} d_{TV}(P^t(v,\cdot),\pi) \le 1/4\}$, where d_{TV} is the total variation distance; see e.g. (Paulin, 2015).

Proposition 4.7. *Suppose that the arrival process is Markov modulated with a chain M^t that has invariant measure π and mixing time t_{mix}. Then, if both M^T and $\widehat{M}^T$ start at the same state, Eq. (5) is satisfied with $\theta_1 = 4n$ and $\theta_2 = 48\beta^2/t_{mix}$, hence the regret is at most $\mathcal{O}(dr_{\max}n\kappa^2 t_{mix}/\beta^2)$. Furthermore, the chain can be hidden from the controller.*

5. Numerical Experiments

We now present numerical experiments to emphasize the following: (i) a single trace suffices for bounded regret, (ii) the importance and usefulness—beyond our algorithm—of using *action scores* as a tie-breaking rule for estimated value functions, and (iii) the effectiveness of our approach in settings beyond those covered by Theorem 2.2. We use the multi-secretary problem (Example 2.1) for the first two objectives, and online packing and matching with reusable resources (Section 2.2) for the third.

5.1. Multi-Secretary with Single Trace

We simulate an instance with $n = 4$ types, and non-stationary arrival process (in particular, we choose p_j^t to be sinusoidal with type-dependent shift, frequency, and translation). The base instance has horizon of length $T = 100$ and initial resource stock of $S^T = 60$; we then scale this to get a family of instances, where the k-th instance has horizon kT and initial resource stock kS^T. Fig. 1 compares different algorithms—RABBI, empirical value-function approximation (DP) with randomized tie-breaking and score-based tie-breaking, and the minimax optimal policy (DJSW) (Devanur et al., 2019)—for this setting.

Fig. 1b shows that RABBI achieves almost 100% of the offline reward. Moreover, we also see that DP with a random tie-breaking rule has regret linear in T, but if we instead use the score-based tie-breaking rule, then it has the same performance as RABBI (thereby confirming Theorem 2.3). We note though that it is not feasible to run DP in higher dimensions, while RABBI is always efficient.

We also benchmark RABBI against the minimax optimal algorithm for this setting (Devanur et al., 2019), which we indicate as DJSW. This policy has a tuning parameter ε, and we search over this parameter and report the best performance. While DJSW does not naturally incorporate historical traces, we also test a modified version where we first feed DJSW the historical trace for T periods (but ignore its actions), and then the online trace for the next T periods. Both algorithms perform much worse, though we note that this is expected since they are designed for maximizing worst-case performance.

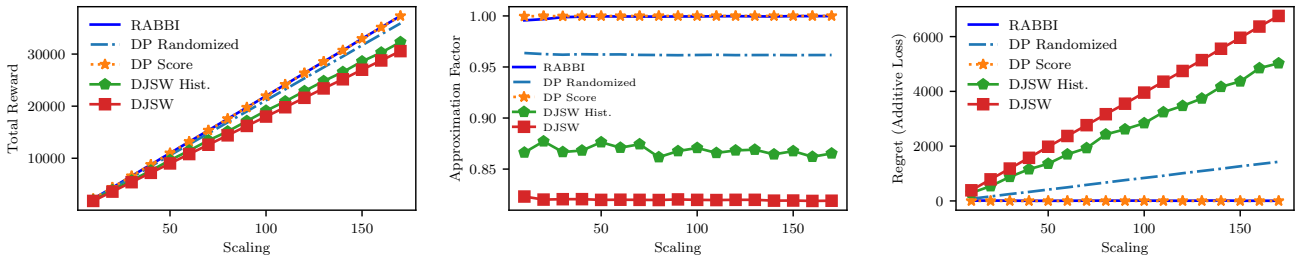
5.2. Online Matching with Reusable Resources

Next we consider online matching with reusable resources (cf. Section 2.2): A type-j customer generates reward $r_{ij} \ge 0$ if assigned a resource $i \in [d]$, and utilises i for $l_j \ge 1$ periods, after which it is returned to the platform. The offline problem at time t is thus $\max_{\mathbf{y}\in\mathbb{R}_{\ge 0}^{n\times[t]}} \sum_{i,j} r_{ij}y_{ij}^t$ subject to natural per-period matching constraints.

In Fig. 2, we compare the performance of RABBI to the state-of-the-art approach (ADP with tuned linear basis functions) for this setting (Rusmevichientong et al., 2020). For fixed T, we study two instances, with stationary (Fig. 2a) and with time-varying (Fig. 2b) arrival processes, and report the regret as a function of number of traces (with 'inf' denoting full model specification). Additionally, for the latter instance, we also demonstrate the performance (reward) as we scale the horizon T (Fig. 2b). Our experiments show that RABBI with a single trace, beats the specialized algorithm *even if it is provided the full distribution*.

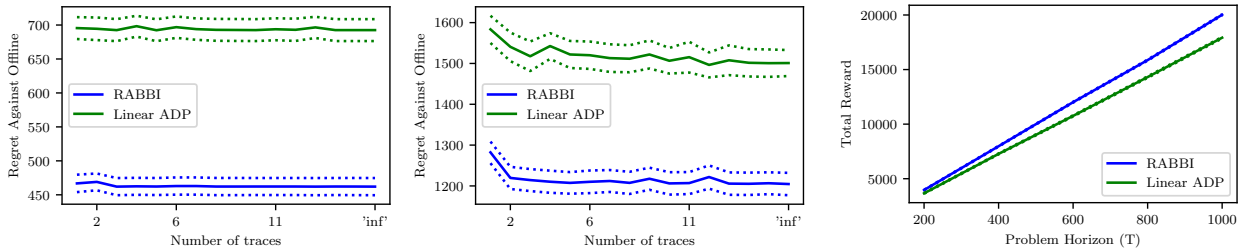
5.3. Online Packing with Reusable Resources

Finally we study online packing with reusable resources (Section 2.2): A type-j customer offers reward r_j for using resources $\{i \in [d] : A_{ij} = 1\}$ for $l_j \ge 1$ periods. The period-t offline problem encodes the natural packing constraints (omitted due to space). For this setting we are unaware of any specialized algorithm (with the exception of the single-resource, infinite horizon case (Levi & Radovanović, 2010)), and so we compare only to the offline benchmark.



(a) Total reward vs. problem size (b) Approximation factor vs. problem size (c) Regret vs. problem size scaling

Figure 1. Performance of different policies for the multi-secretary problem: The plots show the total reward and approximation factor for different algorithms as the size of the problem instance scales. We compare RABBI against (i) the DP with empirical value-functions and randomized tie-breaking, (ii) the DP with empirical value-functions with RABBI's score-based tie breaking, and (iii) two variants of the online learning policy of (Devanur et al., 2019), which we refer to here as DJSW. All algorithms are given a single historical trace.



(a) RABBI vs ADP for different matching instances and increasing traces: Stationary arrivals (left) and time-varying arrivals (right).

(b) Performance (total reward) vs. increasing horizon for RABBI and ADP

Figure 2. Performance of RABBI in the reusable matching problem. We compare RABBI against a specialised algorithm based on Approximate DP with Linear Basis Functions. We give both algorithms an increasing number of historical traces as well as the full arrival model (denoted as 'inf' in plots (a) and (b)). Dotted lines represent 90% confidence intervals.

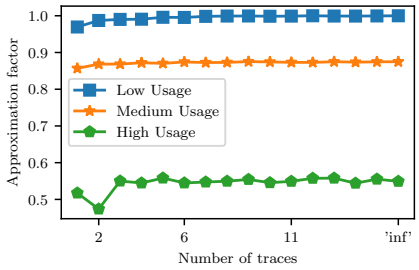


Figure 3. Performance of RABBI in different instances of reusable packing with increasing lengths l_j (number of periods that requests use the resources). In the x-axis we denote as 'inf' the case where we give RABBI the full distribution.

We consider instances with horizon $T = 200$, and test RABBI in 3 different instances: a low-usage instance with $d = 2$ resources, $n = 6$ request types, and random holding times ℓ_j with average 3; a medium-usage instance with $d = 4$, $n = 7$, and average holding-time 6; and a high-usage instance with $d = 7$, $n = 5$, and average holding-time 10. We report results in Fig. 3. Models with larger holding-times naturally lead to harder problems, as supported by our experiments. Moreover, additional traces leads to small performance improvements, as is observed in the reusable matching and multi-secretary settings.

6. Proofs of Main Results

PROOF OF THEOREM 4.2. Due to space constraints, we highlight the main novel aspects of adapting the RABBI algorithm from (Vera et al., 2019) to work with traces, and defer the reader to that work for specifics of the Bellman Inequalities framework. In particular, using Theorem 5 from (Vera et al., 2019) we can bound the regret of RABBI in terms of the information loss and Bellman loss, as $\sum_{t=1}^{T}(dr_{\max}\mathbb{P}[\mathcal{Q}(t,S^t)] + \mathbb{E}[L_B(t,S^t)])$, where S^t denotes the state of RABBI, i.e., the process that evolves according to using controls given by RABBI; $\mathcal{Q}(t,s)$ is the disagreement set, which corresponds to the set of online traces where the control chosen by RABBI is not optimal for offline given the same state; and $L_B(t,s)$ is the Bellman loss, which corresponds to the violation of $\widehat{\varphi}$ in Eq. (4) w.r.t. the Bellman equations.

Using assumptions 1 and 4, we conclude from (Vera et al., 2019, Proposition 4 (Appendix A)) that the Bellman loss is

bounded by $dr_{\max}$ since the functions h and g satisfy both assumptions required in that result. Moreover, we claim that $\mathbb{P}[\mathcal{Q}(t,S^t)] \leq \mathbb{P}[||Z^t - \widehat{Z}^t|| \geq \widehat{Z}^t_j/|\mathcal{U}|\kappa]$, where $j = \xi^t$ is the current arrival. Assuming this claim and using Eq. (5), we obtain $\sum_{t=1}^T \mathbb{P}[\mathcal{Q}(t,S^t)] \leq \sum_{t=1}^T \theta_1 e^{-\frac{\theta_2 t}{(|\mathcal{U}|\kappa)^2}}$, proving the final regret bound.

To prove the claim we use the Lipschitz continuity of LPs as discussed in Section 4. Fix a time t and let $\widehat{X}$ be the solution to the optimization problem in Eq. (4). There exists a solution X to the same optimization problem but with Z^t in lieu of $\hat{Z}^t$ such that $||X - \widehat{X}||_\infty \leq \kappa||Z^t - \widehat{Z}^t||_\infty$. We remark that X corresponds to the problem solved by the offline controller. Let u be the control chosen by RABBI. This control is not optimal for offline only if $X_{uj} < 1$, which formally is $\mathcal{Q}(t,S^t) = \{\omega \in \Omega : X_{uj} < 1\}$. Intuitively, the control u is not optimal only if the offline problem never uses the control u for type j ($X_{uj} < 1$).

Since RABBI chooses the control u with maximum entry and we have the constraint $\sum_{u' \in \mathcal{U}} \widehat{X}_{u'j} = Z^t_j$, necessarily $\widehat{X}_{uj} \geq \frac{Z^t_j}{|\mathcal{U}|}$. In conclusion $X_{uj} < 1$ necessitates $||X - \widehat{X}||_\infty \geq \frac{Z^t_j}{|\mathcal{U}|}$. Applying the Lipshichtz property, we have $\mathbb{P}[||X - \widehat{X}||_\infty \geq \frac{Z^t_j}{|\mathcal{U}|}] \leq \mathbb{P}[||Z^t - \widehat{Z}^t||_\infty \geq \frac{Z^t_j}{\kappa|\mathcal{U}|}]$ concluding the proof of the claim. □

PROOF OF THEOREM 2.3. With a single trace, DP has the following description. At time t, DP solves an *integer program* that corresponds to Eq. (4) with integrality constraints. Let X be the solution of said IP. Then, if $\xi^t = j$, the control u is a maximizer of the Bellman equation if $X_{uj} \geq 1$. Observe that *there could be multiple such controls*. If in case of ties DP uses RABBI's decision rule, i.e., play u with maximum X_{uj}, then the analysis is exactly the same as in the proof of Theorem 2.2 except that we use the Lipschitz property of integer programs (Cook et al., 1986). □

PROOF OF PROPOSITION 4.3. For brevity, all the norms $||\cdot||$ denote the infinity norm. We prove that $\mathbb{P}[||Z^t - \widehat{Z}^t|| \geq \widehat{Z}^t_k/c] \leq 4ne^{-\frac{\beta^2 t}{3(2c+1)^2}}$. Standard concentration results imply that, for $x < \min_j \mu^t_j$,

$$\mathbb{P}[||Z^t - \mu^t|| \geq x] \leq 2\sum_j e^{-\frac{x^2}{3\mu^t_j}}. \tag{6}$$

Call E_x the event where $||Z^t - \mu^t|| < x$ and $||\widehat{Z}^t - \mu^t|| < x$. We use total probabilities to compute:

$$\mathbb{P}\left[||Z^t - \widehat{Z}^t|| \geq \frac{\widehat{Z}^t_k}{c}\right] \leq \mathbb{P}\left[||Z^t - \widehat{Z}^t|| \geq \frac{\widehat{Z}^t_k}{c}|E_x\right] + \mathbb{P}[E^c_x] \leq \mathbb{1}_{\{x+x > (\mu^t_k - x)/c\}} + \mathbb{P}[E^c_x].$$

The last inequality is because, in E_x, $\widehat{Z}^t_k > \mu^t_k - x$. Set $x = \frac{\min_j \mu^t_j}{2c+1}$ and observe that, with this choice, the indicator is zero and we satisfy the requirements of Eq. (6). Further, by our condition we have $x \geq \frac{\beta t}{2c+1}$ and it always holds that $\mu^t_j \leq t$, hence

$$\mathbb{P}[||Z^t - \widehat{Z}^t||_\infty \geq \widehat{Z}^t_k/c] \leq \mathbb{P}[E^c_x] \leq 4\sum_j e^{-\frac{\beta^2 t}{3(2c+1)^2}}.$$

This concludes the proof. □

PROOF OF PROPOSITION 4.5. Observe that the crucial fact in the proof of Proposition 4.3 is that both Z^t and $\widehat{Z}^t$ concentrate around μ^t, see Eq. (6), where μ^t does not depend on the realization, i.e., it is fixed for all t. We prove a similar fact, but with another quantity that is also common to both traces. We claim that

$$\mathbb{P}[||Z^t - t\pi(p)||_\infty \geq x] \leq 4ne^{-\frac{1-\lambda}{12t}x^2}, \tag{7}$$

where $\pi(p)_j := \sum_{v \in V} \pi(v) p_j(v)$. Assuming Eq. (7), then the proof proceeds exactly as in Proposition 4.3.

What remains is to prove Eq. (7). Conditioned on $M^t = v^t, \ldots, M^1 = v^1$, the arrivals of type j are independent with probabilities $p_j(v^\tau)$ for $\tau = t, \ldots, 1$. Denoting $p_j(v^{t:1}) = p_j(v^t) + \ldots + p_j(v^1)$ and $E(v^{t:1})$ the event $M^t = v^t \ldots, M^1 = v^1$, let us define the following r.v. conditioned on M^t:

$$\eta^t_j = \sum_{v^t, \ldots, v^1} \mathbb{1}_{E(v^{t:1})}(p_j(M^t) + p_j(v^{t-1:1})). \tag{8}$$

Then, $\mathbb{P}[|Z^t_j - \eta^t_j| \geq x|M^t]$ is upper bounded by

$$\sum_{v^t, \ldots, v^1} \mathbb{P}[E(v^{t:1})|M^t]\mathbb{P}\left[\left|Z^t_j - \frac{p_j(l^{t:1})}{t}\right| \geq x|E(v^{t:1})\right]$$
$$\leq \sum_{v^t, \ldots, v^1} \mathbb{P}[E(v^{t:1})|M^t] 2e^{-\frac{x^2}{3t}} = 2e^{-\frac{x^2}{3t}}. \tag{9}$$

The inequality follows from Eq. (6).

We know that Z^t_j concentrates around η^t_j and $\widehat{Z}^t_j$ concentrates around $\widehat{\eta}^t_j$ (as defined in Eq. (8), but with $\widehat{M}^t$ replacing M^t). Now we will show that both η^t_j and $\widehat{\eta}^t_j$ concentrate around $t\pi(p)_j$. Indeed, applying (Fan et al., 2018, Theorem 2.1), provided that $M^T \sim \pi$, we obtain the following

$$\mathbb{P}\left[\left|\eta^t_j - t\pi(p_j)\right| > x\right] \leq 2e^{-2\frac{1-\lambda}{1+\lambda}\frac{x^2}{t}}. \tag{10}$$

We use $|Z^t_j - t\pi(p)_j| \leq |Z^t_j - \eta^t_j| + |\eta^t_j - t\pi(p)_j|$ and evaluate Eqs. (9) and (10) with $x/2$ to obtain

$$\mathbb{P}[|Z^t_j - t\pi(p)_j| \geq x] \leq 2e^{-\frac{x^2}{12t}} + 2e^{-\frac{1-\lambda}{1+\lambda}\frac{x^2}{2t}}.$$

A union bound finishes the proof of Eq. (7). □

PROOF OF PROPOSITION 4.7. We reason as in the proof of Proposition 4.5, but in this case we prove that Z^t and $\widehat{Z}^t$ concentrate around $\mathbb{E}[\eta^t]$ which is also invariant of the

trace (since both chains start at the same state). Formally, we claim $\mathbb{P}[||Z^t - \mathbb{E}[\eta^t]||_\infty \geq x] \leq 4ne^{-\frac{x^2}{12t t_{\text{mix}}}}$.

To prove the claim, we use $|Z_j^t - \mathbb{E}[\eta_j^t]| \leq |Z_j^t - \eta_j^t| + |\eta_j^t - \mathbb{E}[\eta_j^t]|$ so that

$$\mathbb{P}[|Z_j^t - \mathbb{E}[\eta_j^t]| \geq x] \leq \mathbb{P}\Big[|Z_j^t - \eta_j^t| \geq \frac{x}{2} \text{ or } |\eta_j^t - \mathbb{E}[\eta_j^t]| \geq \frac{x}{2}\Big].$$

The first event is bounded by Eq. (9) whereas for the second we apply (Paulin, 2015, Corollary 2.10) to obtain $\mathbb{P}[|\eta_j^t - \mathbb{E}[\eta_j^t]| \geq x] \leq 2e^{-2\frac{x^2}{t t_{\text{mix}}}}$. Finally, a union bound finishes the proof of the claim. □

References

Agrawal, S. and Devanur, N. R. Bandits with concave rewards and convex knapsacks. In *Proceedings of the fifteenth ACM conference on Economics and computation*, pp. 989–1006. ACM, 2014.

Agrawal, S., Wang, Z., and Ye, Y. A dynamic near-optimal algorithm for online linear programming. *Operations Research*, 62(4):876–890, 2014.

Arlotto, A. and Gurvich, I. Uniformly bounded regret in the multisecretary problem. *Stochastic Systems*, 2019.

Azar, P. D., Kleinberg, R., and Weinberg, S. M. Prophet inequalities with limited information. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pp. 1358–1377. SIAM, 2014.

Badanidiyuru, A., Kleinberg, R., and Slivkins, A. Bandits with knapsacks. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pp. 207–216. IEEE, 2013.

Buchbinder, N., Naor, J. S., et al. The design of competitive online algorithms via a primal–dual approach. *Foundations and Trends® in Theoretical Computer Science*, 3 (2–3):93–263, 2009.

Bumpensanti, P. and Wang, H. A re-solving heuristic with uniformly bounded loss for network revenue management. *arXiv preprint arXiv:1802.06192*, 2018.

Cook, W., Gerards, A. M., Schrijver, A., and Tardos, É. Sensitivity theorems in integer linear programming. *Mathematical Programming*, 34(3):251–264, 1986.

Devanur, N. R., Jain, K., Sivan, B., and Wilkens, C. A. Near optimal online algorithms and fast approximation algorithms for resource allocation problems. *Journal of the ACM (JACM)*, 66(1):7, 2019.

Fan, J., Jiang, B., and Sun, Q. Hoeffding's lemma for markov chains and its applications to statistical learning. *arXiv preprint arXiv:1802.00211*, 2018.

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

Jasin, S. and Kumar, S. A re-solving heuristic with bounded revenue loss for network revenue management with customer choice. *Mathematics of Operations Research*, 37 (2):313–345, 2012.

Kleinberg, R. and Weinberg, S. M. Matroid prophet inequalities. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pp. 123–136. ACM, 2012.

Levi, R. and Radovanović, A. Provably near-optimal lp-based policies for revenue management in systems with reusable resources. *Operations Research*, 58(2):503–507, 2010.

Mangasarian, O. and Shiau, T. Lipschitz Continuity of Solutions of Linear Inequalities, Programs and Complementarity Problems. *SIAM Journal on Control and Optimization*, 25(3):583–595, 1987.

Mannor, S., Simester, D., Sun, P., and Tsitsiklis, J. N. Bias and variance approximation in value function estimates. *Management Science*, 53(2):308–322, 2007.

Paulin, D. Concentration inequalities for markov chains by marton couplings and spectral methods. *Electronic Journal of Probability*, 20, 2015.

Rubinstein, A., Wang, J. Z., and Weinberg, S. M. Optimal single-choice prophet inequalities from samples. In *11th Innovations in Theoretical Computer Science Conference (ITCS 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.

Rusmevichientong, P., Sumida, M., and Topaloglu, H. Dynamic assortment optimization for reusable products with random usage durations. *Management Science*, 2020. Forthcoming.

Vera, A. and Banerjee, S. The bayesian prophet: A low-regret framework for online decision making. In *Abstracts of the 2019 SIGMETRICS/Performance Joint International Conference on Measurement and Modeling of Computer Systems*, pp. 81–82. ACM, 2019.

Vera, A., Banerjee, S., and Gurvich, I. Online allocation and pricing: Constant regret via bellman inequalities. *arXiv preprint arXiv:1906.06361*, 2019.