Operations Research

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

Technical Note—Near-Optimal Bayesian Online Assortment of Reusable Resources

Yiding Feng, Rad Niazadeh, Amin Saberi

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

OPERATIONS RESEARCH
Vol. 72, No. 5, September–October 2024, pp. 1861–1873
ISSN 0030-364X (print), ISSN 1526-5463 (online)

Crosscutting Areas

Technical Note—Near-Optimal Bayesian Online Assortment of Reusable Resources

Yiding Feng,[a] Rad Niazadeh,[a,*] Amin Saberi[b]

[a] University of Chicago Booth School of Business, Chicago, Illinois 60637; [b] Management Science and Engineering, Stanford University, Stanford, California 94305

*Corresponding author

Contact: yiding.feng@chicagobooth.edu, https://orcid.org/0000-0002-8258-6994 (YF); rad.niazadeh@chicagobooth.edu, https://orcid.org/0000-0002-5880-6221 (RN); saberi@stanford.edu (AS)

Received: October 20, 2020
Revised: April 9, 2022; April 17, 2023
Accepted: February 7, 2024
Published Online in Articles in Advance: July 15, 2024

Area of Review: Market Analytics and Revenue Management

https://doi.org/10.1287/opre.2020.0687

Abstract. Motivated by the applications of rental services in e-commerce, we consider revenue maximization in online assortment of reusable resources for a stream of arriving consumers with different types. We design competitive online algorithms with respect to the optimum online policy in the Bayesian setting in which types are drawn independently from known heterogeneous distributions over time. In the regime where the minimum of initial inventories $c_{\min}$ is large, our main result is a near-optimal $1 - \min\left(\frac{1}{2}, \sqrt{\log(c_{\min})/c_{\min}}\right)$ competitive algorithm for the general case of reusable resources. Our algorithm relies on an expected LP benchmark for the problem, solves this LP, and simulates the solution through an independent randomized rounding. The main challenge is obtaining point-wise inventory feasibility in a computationally efficient fashion from these simulation-based algorithms. To this end, we use several technical ingredients to design *discarding policies*—one for each resource. These policies handle the trade-off between the inventory feasibility under reusability and the revenue loss of each of the resources. However, discarding a unit of a resource changes the future consumption of other resources. To handle this new challenge, we also introduce *postprocessing* assortment procedures that help with designing and analyzing our discarding policies as they run in parallel, which might be of independent interest. As a side result, by leveraging techniques from the literature on prophet inequality, we further show an improved near-optimal $1 - 1/\sqrt{c_{\min} + 3}$ competitive algorithm for the special case of nonreusable resources. We finally evaluate the performance of our algorithms using the numerical simulations on the synthetic data.

Funding: R. Niazadeh's research is partially supported by an Asness Junior Faculty Fellowship from the University of Chicago Booth School of Business.
Supplemental Material: The online appendix is available at https://doi.org/10.1287/opre.2020.0687.

Keywords: analysis of algorithms • analysis of algorithms • suboptimal algorithms • inventory/production • approximations/heuristics

1. Introduction

Assortment planning refers to the decision of a revenue-maximizing firm as to which subset of products to display to its consumers. In classic retail applications, the focus is mostly on sale; however, with the advent of online e-commerce platforms, several applications have emerged where the focus is on renting out *reusable resources*. A reusable resource—also referred to as a rental product—leaves the stock for some time duration after being assigned to a consumer and can be reassigned to a new consumer once it is back. Examples are virtual machines in cloud computing platforms such as AWS, houses in vacation rental online marketplaces such as Airbnb, and local professional services in online labor platforms such as Thumbtack. In order to extract more revenue, a (personalized) assortment policy can decide to display a different subset of these resources once a new consumer interacts with the platform; the task of such a policy is to manage the sequence of assortments in the long run given the inventory restrictions.

Motivated by the applications listed, we study the online assortment of reusable resources in which a platform sequentially makes irrevocable assortment decisions for a stream of arriving consumers. All consumers have a *type* that determines their choice probabilities given each possible assortment—referred to as the consumer's choice model. The platform collects a one-time personalized payment each time a consumer rents a product. Different products have different rental fees that are determined by the consumer's type. To model the uncertainty about the duration of a rental, we consider a stochastic model where the rental durations are drawn independently each time a product is rented. The types of arriving consumer also determine their rental duration distributions for different products. To model the platform's prior information about its future consumers, which is usually formed based on the past consumers' data in an online platform, we take a Bayesian approach. We assume the types are drawn independently over time from *known heterogeneous distributions*.

Once new consumers arrive, the platforms observe their realized type from the known type distribution, display a new subset of products, and allow the consumers to select a product stochastically from this subset based on their choice models. The goal is to design an online assortment algorithm in order to maximize the expected total collected rental fees (also known as the revenue) during the decision-making horizon. Importantly, we consider the setting where each product has an initial inventory and the algorithm should always assort an available subset of products. Availabilities are determined by the current inventory levels of the products—quantities that decrease as a unit of the product is rented and increase as it is returned to the stock.

Given the sequence of distributions for the entire decision-making horizon, a *revenue benchmark* in our problem is an upper bound on the expected revenue of any feasible online algorithm as the platform, where the expectation is over the randomness in the algorithm, the type sequence, and the consumers' choices. We measure the performance of our online algorithms using the notion of *competitive ratio*, that is, the worst-case ratio between the expected revenue of the online algorithm and the targeted revenue benchmark.

As is common in the literature and practice of assortment optimization, we assume general consumer choice models that are weak substitutes; that is, assorting a new product only weakly decreases the choice probability of another assorted product (see Assumption 1). With no further assumptions, even the one-shot assortment optimization can be computationally hard. To resolve this issue, we assume having oracle access to a black-box algorithm that can solve the one-shot assortment optimization (see Assumption 3). Again, this is a common assumption in the literature to get around the computational issue associated with the one-shot problem when general consumer choice models are considered (Golrezaei et al. 2014, Rusmevichientong et al. 2020). In the oracle-access computational model, we aim to design polynomial-time competitive online algorithms with respect to an appropriate revenue benchmark for our problem.

In this setting, the question of finding the optimum online policy that maximizes the expected total revenue is purely computational, and an exponential-size dynamic programming (DP) can formulate the optimum online. Whereas no formal computational hardness is known for computing the optimum online in our problem, it is conjectured to be computationally hard, even in the oracle-access model.[1] Therefore, it is natural to consider the expected revenue of the optimum online policy as a revenue benchmark and study whether it is amenable to polynomial time competitive online algorithms. More specifically, we ask the following question in this paper:

> *How close a polynomial time online algorithm can be to the optimum online policy in terms of expected revenue? In particular, can we obtain constant or near-optimal competitive online algorithms with respect to the optimum online policy (when initial inventories are large)?*

A significant progress toward providing a compelling answer to this question is the result of Rusmevichientong et al. (2020), which establishes an approximate dynamic programming approach for the exponential-size optimum online DP. They show that a greedy algorithm that uses a linear approximation of the optimal revenue to go function obtains at least $\frac{1}{2}$ of the expected revenue of the optimum online. They further study the setting when rental durations are infinite (i.e., the resources are not reusable) and rental fees are type independent. In this setting, they show how to perform a rollout on a simple static policy to obtain $1-\min\left(\frac{1}{2}, \frac{1}{\sqrt[3]{c_{\min}}}\right)$ fraction of the expected revenue of the optimum online policy, where $c_{\min}$ is the minimum initial inventory across different products.[2] This last result is essentially a near-optimal competitive algorithm when the initial inventories are large and the resources are nonreusable. Note that the large initial inventory regime is relevant in many applications of assortment optimization, and it is the main focus of our paper as well.

1.1. Our Contributions

The main contribution of our paper is the following result.

Main Result. *For the general case of Bayesian online assortment of reusable resources, we propose a polynomial time online algorithm that obtains a near-optimal competitive ratio of* $1-\min\left(\frac{1}{2}, \varepsilon^*(c_{\min})\right)$, *where* $\varepsilon^*(x) = O(\sqrt{\log(x)/x})$.

This competitive ratio guarantee holds even when the rental fees, consumer choices, and rental duration distributions are type dependent and vary arbitrarily across different types.

To obtain the result given, our work diverges from Rusmevichientong et al. (2020) by considering a different revenue benchmark. In particular, we consider a linear programming relaxation of the optimum online policy, referred to as the *Bayesian expected LP*. To define this benchmark, suppose a feasible online policy knows the exact realizations of future types but does not know the realizations of consumer choices and rental durations. The optimum such policy, known as the *clairvoyant optimum online*, clearly provides a revenue benchmark. Now consider a relaxation of this policy by only requiring the inventory feasibility constraints of reusable resources to hold in expectation over the randomness in types, consumer choices, and rental durations. Given the sequence of type distributions, this relaxation is encoded by an LP with an exponential number of variables and polynomial number of time-varying packing constraints to ensure the inventory feasibility of reusable resources in expectation. See Section 2.2 for details.

It turns out that we can simply solve the Bayesian expected LP in polynomial time by solving its dual program using the ellipsoid method—given access to the offline assortment oracle. Given the LP solution, a simple but powerful technique in the Bayesian online optimization is to use a *simulation-based* rounding algorithm to mimic the optimal solution of this LP (for examples of this approach, refer to Alaei et al. 2012, Devanur et al. 2012, Gallego et al. 2015, Dickerson et al. 2018, Wang et al. 2018, Ma et al. 2020, Baek and Ma 2022). After observing the type of arriving consumer, this algorithm independently samples an assortment from a distribution over subsets of products that comes from the LP solution, ignoring the inventory constraints. This algorithm has no loss in terms of the expected revenue compared with the LP solution; however, it only respects the inventory constraints of each product in expectation—and not necessarily under every sample path of the existing randomness.

Our main technical contribution is providing techniques to transform the simulation-based algorithm into a point-wise feasible online algorithm in polynomial time, with constant or negligible multiplicative loss in the expected revenue. To this end, we run a separate procedure over time—one for each product—together with the simulation-based policy. After an assortment is sampled at each time, each procedure decides whether to *discard* the corresponding product if it is in the sampled subset to maintain the inventory feasibility of this product. This specific architecture of sampling according to the LP solution and then using product-specific discarding rules has been explored in the past; for example, see Alaei et al. (2012), Gallego et al. (2015), Dickerson et al. (2018), Wang et al. (2018), and Baek and Ma (2022).[3] Similarly, we aim to design polynomial time online discarding policies (and other necessary algorithmic constructs) that handle the trade-off between maintaining the inventory feasibility and the discarding revenue loss for each product. The main new challenges specific to our problem is that (i) products are reusable, (ii) products are weak substitutes—and hence discarding a product increases the choice probability of other products, and (iii) discarding can be potentially randomized—which combined with reusability creates complicated correlation structures among selection indicator random variables across time and makes it challenging to argue about point-wise feasibility of policies. In what follows, we sketch our main technical contributions and how they overcome these challenges.

1.1.1. General Rental Duration Distributions/Near-Optimal Discarding (Section 3.2). The main idea behind our near-optimal discarding procedure is discarding each available sampled product independently at random with a small probability. This is a simple yet reasonable approach (cf. Hajiaghayi et al. 2007), as the simulation-based algorithm respects the inventory constraints of each product in expectation. By independent randomized discarding with probability $\gamma > 0$, we leave some slack in the inventory feasibility constraint by ensuring that the expected value of the number of units of the product under rental is at most $(1-\gamma)$ times its initial inventory amount at any time. If this quantity as a sum of independent rental indicator random variables concentrates around its expectation, we will then avoid violation of the inventory constraint with high probability when $\gamma = O(\sqrt{\log(c_{\min})/c_{\min}})$. Moreover, it only loses γ fraction of the expected revenue form this product, as desired.

However, this simple approach does not work as described because (i) the resources are reusable and the inventories are limited; hence, the rental indicator random variable of a product at some time $\tau < t$ can be correlated with the rental indicator random variable of the same product at time t if the rental duration of time τ is at least $t-\tau$, and the last unit of the product is rented at τ; (ii) once a discarding procedure drops a product from the sampled assortment, there will be less cannibalization of other products, as the consumer choices are weak substitutes. This, in turn, increases the probability of other products being chosen by the arriving consumer and hence increases the expected number of units of different products under rental in future for the resulting algorithm compared with what is expected from the simulation-based algorithm.

We fix these issues by proposing a postprocessing step after the independent randomized discarding, which we refer to as *subassortment sampling*. In a nutshell, the goal of the subassortment sampling is to find a distribution over available subsets so that the products that are not discarded will be rented with *exactly* the same probability as in the optimal solution of the Bayesian expected LP. It is not even clear a priori whether such a distribution exists; nevertheless, we show it does and provide a polynomial time construction to sample from this distribution. Using the properties of the subassortment sampling, we propose a coupling trick to show our desired concentration despite the fact that the rental indicator random variables are correlated across time. Note that the task of subassortment sampling is quite general, and it might be of independent interest in other applications.[4]

1.1.2. General Rental Duration Distributions/$\frac{1}{2}$-Competitive Discarding (Section 3.3). As an alternative discarding policy for the general case of reusable resources, consider an exponential-size DP that keeps track of the state of each unit of the product (i.e., when each unit returns to the inventory) and solves the discarding task optimally. We introduce an approximate version of this DP, which we also refer to as *optimistic DP*, that can be solved in polynomial time. The goal is to maximize the per-unit revenue to go of the product when the inventory is automatically *replenished* by an exogenous process every time we make a discarding decision so that the entire inventory of the product is always available on hand. This new DP is inventory independent and thus is polynomial size. We then consider a discarding algorithm that makes the same decisions as the optimistic DP. The result is a nonadaptive thresholding discarding rule; that is, an available product is only discarded if its rental fee is below a certain threshold. These thresholds are computed up front and only depend on the product, time, and realized type.

The main intuition behind why the discarding algorithm is a reasonable approximation is as follows. We can show the expected revenue to go of this algorithm is a concave function of the inventory level; that is, the higher the inventory level, the lower the per-unit expected revenue to go. As a result, when there is no replenishment in reality, it obtains at least the same expected per-unit revenue to go as the DP with replenishment. We then analyze the worst-case ratio between the value of this inventory-independent DP and the per-unit revenue of the expected LP using a "factor revealing linear program" and its dual. This approach establishes a lower-bound of $\frac{1}{2}$ for this ratio. It is worth noting that similar proof techniques based on dual fitting have been used in the literature for other problems with nonreusable resources (see, e.g., Adelman 2007, Zhang and Adelman 2009, Alaei et al. 2012, Gallego et al. 2015, Wang et al. 2018). Our work extends the existing analysis to prove performance guarantees for LP-based discarding policies when resources are reusable.

We highlight that our DP for designing the approximate discarding policy given the expected LP solution shares similar, but not completely identical, recursive structures with the approximate dynamic programming approach in Rusmevichientong et al. (2020) for directly approximating the optimum online policy; in fact, in contrast to their approach, the Bellman update equation of our DP uses the solution of the expected LP (the optimal assortment sampling probabilities). Therefore, whereas both DPs provide the same approximation factor of $\frac{1}{2}$, ours is with respect to the *stronger* benchmark of expected LP, versus theirs, which is with respect to the optimum online policy. This subtle difference turns out to be the key in combining the performance guarantees of our two algorithms in Sections 3.2 and 3.3 in order to obtain a simple hybrid algorithm that achieves the theoretical *"best of both worlds"* competitive ratio guarantee with respect to the stronger expected LP benchmark (and as we observe later, improved performance in numerical simulations).

1.1.3. Hybrid Simulation-Based Algorithm (Section 3.4). Having access to the aforementioned simulation-based algorithms with different discarding rules, we aim to define a *hybrid algorithm* that enjoys the competitive ratios of both small and large inventory regimes. To this end, we make an upfront decision on which discarding policy to use for each product. In particular, we use the value function of the optimistic DP for each product separately to calculate the ratio $\mathcal{R}_i$ between the expected revenue to go of following the optimistic DP for this product and the contribution of this product to the expected LP's objective. We then compare this ratio with $1-\varepsilon^*(c_i)$ (see (1) for the definition of function $\varepsilon^*(\cdot)$) to partition the products into *large inventory* (i.e., when $\mathcal{R}_i+\varepsilon^*(c_i)<1$) and *small inventory* (i.e., when $\mathcal{R}_i+\varepsilon^*(c_i)>1$). For each large inventory product, we run the randomized discarding policy in Section 3.2, and for each small inventory product, we run the optimistic DP discarding policy in Section 3.3. We also use the subassortment sampling procedure for postprocessing to correct the resulting increase in choice probabilities of nondiscarded products because of weak substitution. By using the facts that (i) the competitive ratio analyses of both algorithms decouple across products, (ii) both analyses compare the expected revenue obtained from each product with the contribution of that product to the expected LP's objective, and (iii) subassortment sampling corrects the choice probabilities of nondiscarded products, we show the resulting hybrid policy combines the two competitive ratios.[5] Whereas this hybrid algorithm attains the best of both worlds competitive ratio of $1-\min\left(\frac{1}{2},\varepsilon^*(c_{\min})\right)$, it is also likely to outperform both policies in practical scenarios; the results of our numerical simulations in Section EC8 in the Online Appendix empirically support this claim. We also present a second hybrid algorithm using the method of conditional expectations. See Section 3.4 for more details.

We also complement our results by considering the special case of nonreusable resources. By leveraging techniques from the literature on prophet inequality and extending them to the Bayesian assortment optimization problem, we provide a near-optimal improved competitive ratio of $\left(1-\frac{1}{\sqrt{c_{\min}+3}}\right)$ with respect to the expected LP in this setting. See Section EC3 and Section EC7 in the Online Appendix for more details.

1.1.3.1. Numerical Simulations (Online Appendix Section EC8). We finally provide numerical justification for the revenue performance of our proposed policies. Adapting the setups of the numerical experiments in Golrezaei et al. (2014) and Rusmevichientong et al. (2020) to our setting, we compare the revenue of our proposed policies—that is, the hybrid algorithm and the simulation with optimal discarding under infinite rental durations—with other policies in the literature. In our numerical simulations, we consider various scenarios with both general rental duration distributions and infinite rental durations. In all of these scenarios, our policies noticeably outperform the other policies in terms of the expected revenue.

2. Preliminaries

We first formalize our problem, the model, and all the required assumptions in Section 2.1. We then briefly explain various aspects of our expected LP benchmark in Section 2.2.

2.1. Model and Problem Definition

The platform offers n different rental products, indexed by $[n]=\{1,2,\ldots,n\}$. Each rental product i has an initial inventory of $c_i \in \mathbb{Z}_+$. Consumers who are interested in renting these products arrive sequentially at times $t=1,2,\ldots,T$. Consumer t has type $z_t \in \mathcal{Z}_t$, where $\mathcal{Z}_t$ denotes the (discrete) space of possible types at time t. We assume types are drawn independently from *known* probability distributions $F_t : \mathcal{Z}_t \to [0,1]$ at times $t=1,\ldots,T$.

Upon the arrival of consumer t, that consumer's type z_t is revealed to the platform. Given this type and the history up to time t, the platform offers an assortment of available products $S_t \in \mathcal{S}$ from its inventory, where $\mathcal{S} \subseteq 2^{[n]}$ is the collection of all feasible assortments that can be offered, ignoring the inventory availability. Given the assortment S_t, the consumer chooses a rental product $i_t \in S_t$, pays a rental fee to the platform, and keeps the product for a stochastic rental duration $d_t \in \mathbb{Z}_+$.

We consider the setting where the consumer choice behavior, rental fees of different products, and rental duration distributions of different products depend on the type z_t at each time t. Formally, a consumer type z is defined as a tuple $\langle \phi^z, \mathbf{r}^z, \mathbf{G}^z \rangle$ so that

- The choice of a consumer with type z is modeled by a general choice model function $\phi^z : \mathcal{S} \times [n] \to [0,1]$, where $\phi^z(S,i)$ is the probability that consumer with type z chooses product i to rent when assortment set $S \in \mathcal{S}$ is offered.
- For a consumer with type z, $\mathbf{r}^z=(r_1^z, r_2^z,\ldots,r_n^z) \in \mathbb{R}^n$, where r_i^z denotes the rental fee of product i. Moreover, $\mathbf{G}^z=(G_1^z, G_2^z,\ldots,G_n^z)$, where G_i^z denotes the cumulative distribution function (cdf) of rental duration of product i for type z. We use $g_i^z : [T] \to [0,1]$ to denote the probability distribution function (pdf) of rental duration of product i for type z. Moreover, let $\overline{G}_i^z(\cdot) \triangleq 1-G_i^z(\cdot)$.

Note that we assume rental durations are independent across time; that is, if, at time t, a consumer of type z chooses a product i, a fresh sample $d_t \sim G_i^z$ is realized as the rental duration of this product. We further impose the following assumptions on our choice models and feasible assortments, which are common in previous literature (cf. Golrezaei et al. 2014, Rusmevichientong et al. 2014):

Assumption 1 (Weak Substitutability). *For all $t \in [T], z \in \mathcal{Z}_t$, and $i \in [n]$, $\phi^z(\emptyset, i)=0$. Moreover, for all $S \in \mathcal{S}$ and $j \in [n]/\{i\}$, $\phi^z(S,i) \geq \phi^z(S \cup \{j\}, i)$.*

Assumption 2 (Downward-Closed Feasibility). *If $S \in \mathcal{S}$ and $S' \subseteq S$, then $S' \in \mathcal{S}$; that is, a feasible assortment will remain feasible after removing any subset of its offered products.*

Remark 1. In online hospitality services such as Airbnb, users report the duration of their stay to the platform before the platform shows them a listing. In such a variation, the platform makes the assortment decision by using the exact realizations of current rental times for the arriving type. Indeed, this is a special case of our model where the rental time distributions are point mass.

Given type distributions $\{F_t\}_{t=1}^T$, the goal is to design online algorithms—playing the role of the platform—that maximize the expected revenue granted from rental fees; here, the expectation is over randomness of the algorithm (if randomized) and the environment, that is, types, consumer choices, and rental durations. A revenue benchmark for this problem is defined to be any upper bound on the expected revenue obtained by any feasible online algorithm (which might or might not be achievable by a feasible online algorithm). Fixing a revenue benchmark, we evaluate the performance of any online algorithm by its *competitive ratio* against this benchmark. Informally speaking, competitive ratio is the worst-case ratio between the expected total revenue of the online algorithm and the benchmark, where the worst case is over all possible type distributions.

Definition 1 (Competitive Ratio). An online algorithm $\mathcal{A}$ is *α-competitive* against a given revenue benchmark if

$$\inf_{T\geq 1} \inf_{\{F_t\}_{t=1}^T} \frac{\mathbf{Rev}_{\mathcal{A}}[\{F_t\}_{t=1}^T]}{\mathbf{OPT}[\{F_t\}_{t=1}^T]} \geq \alpha,$$

where $\mathbf{Rev}_{\mathcal{A}}[\cdot]$ is the expected revenue of algorithm $\mathcal{A}$, and $\mathbf{OPT}[\cdot]$ is the given revenue benchmark.

For a general consumer choice model, the exact or even approximate offline assortment optimization can be computationally hard (Kök et al. 2008). In order to avoid this obstacle when designing polynomial time online algorithms for general consumer choice models, we assume having access to an algorithm that solves the offline assortment problem. For simplicity, we assume the solver is exact throughout the paper, but all of our results still hold with a multiplicative degrade of β in the competitive ratios if the solver is a β-approximation algorithm for some $0 < \beta < 1$.

Assumption 3 (Offline Oracle). *For all $t \in [T], z \in \mathcal{Z}_t$, and $\hat{\mathbf{R}} \in \mathbb{R}^n_+$, we have oracle access to an algorithm that finds a subset $\hat{S} \in \mathcal{S}$ such that*

$$\hat{S} \in \operatorname*{argmax}_{S \in \mathcal{S}} \sum_{i=1}^{n} \hat{R}_i \phi^z(S, i).$$

2.2. Bayesian Expected LP Benchmark

A key ingredient in all of our algorithms is the *Bayesian expected LP benchmark*—a concept commonly used in previous literature on online allocations, mechanism design, and assortment optimization to remedy issues of the benchmarks given (e.g., see Chawla et al. 2010, Alaei 2014, Gallego et al. 2015, Wang et al. 2018, Anari et al. 2019, Ma et al. 2020). This benchmark, denoted by Expected-LP$[\{F_t\}_{t=1}^T]$, uses linear programming to capture the optimum algorithm that only requires satisfying the inventory constraints in expectation, where expectation is taken over randomness in rental durations and consumer types given type distributions $\{F_t\}_{t=1}^T$:

$$\begin{aligned}
\max_{\mathbf{y} \geq \mathbf{0}} \quad & \sum_{t=1}^{T} \sum_{z_t \in \mathcal{Z}_t} \sum_{S \in \mathcal{S}} \sum_{i=1}^{n} F_t(z_t) r_i^{z_t} \phi^{z_t}(S, i) y_{S,t,z_t} \quad \text{s.t.} \\
& \sum_{\tau=1}^{t} \sum_{z_\tau \in \mathcal{Z}_\tau} \sum_{S \in \mathcal{S}} F_\tau(z_\tau) \overline{G}_i^{z_\tau}(t-\tau) \phi^{z_\tau}(S, i) y_{S,\tau,z_\tau} \leq c_i \qquad i \in [n],\ t \in [T] \\
& \sum_{S \in \mathcal{S}} y_{S,t,z_t} \leq 1 \qquad t \in [T],\ z_t \in \mathcal{Z}_t.
\end{aligned}$$

$$(\text{Expected-LP}[\{F_t\}_{t=1}^T])$$

Here, variables $\{y_{S,t,z_t}\}_{t \in [T], S \in \mathcal{S}, z_t \in \mathcal{Z}_t}$ correspond to probabilities that assortment S is offered to consumer t given type z_t is realized, and first constraint shows inventory feasibility in expectation.

A few explanations are in order. First, the optimal objective value of this LP is an upper bound on the expected revenue of the clairvoyant optimum online benchmark and, hence, the weaker nonclairvoyant optimum online (Proposition 1; see Section EC4 in the Online Appendix for the proof). Second, Expected-LP$[\{F_t\}_{t=1}^T]$ can be solved efficiently using an oracle for the offline assortment (Proposition 2; see Section EC4 in the Online Appendix for the proof). We use this computational block as a preprocessing step in all of our algorithms.[6]

Proposition 1. *For any type distributions $\{F_t\}_{t=1}^T$, the expected total revenue of the clairvoyant optimum online benchmark is upper bounded by Expected-LP$[\{F_t\}_{t=1}^T]$.*

Proposition 2. *Given an algorithm for offline assortment (Assumption 3), an optimal assignment $\{y^*_{S,t,z_t}\}$ of Expected-LP$[\{F_t\}_{t=1}^T]$ can be computed efficiently in time Poly$(n, T, \sum_{t \in [T]} |\mathcal{Z}_t|)$. Moreover, $\{y^*_{S,t,z_t}\}$ has no more than Poly$(n, T, \sum_{t \in [T]} |\mathcal{Z}_t|)$ nonzero entries.*

In Section EC2 in the Online Appendix, we compare the Bayesian expected LP benchmark with other benchmarks considered in the literature.

3. Near-Optimal Algorithm for General Rental Durations

In this section, we present our main result—a near-optimal online simulation-based algorithm with competitive ratio at least $\max\left(\frac{1}{2}, 1 - \varepsilon^*(c_{\min})\right)$ against the Bayesian expected LP benchmark, where

$$\varepsilon^*(x) \triangleq \min_{\gamma \in [0,1]} 1 - (1-\gamma)\left(1 - \exp\left(-\frac{\gamma^2 x}{2-\gamma}\right)\right). \tag{1}$$

Let $\gamma^*(c_{\min})$ be the optimal assignment of γ in Equation (1). It is not hard to verify that $\varepsilon^*(c_{\min}) = O(\sqrt{\log(c_{\min})/c_{\min}})$ and is achieved at $\gamma^*(c_{\min}) = O(\sqrt{\log(c_{\min})/c_{\min}})$. We first sketch our approach in Section 3.1. We then introduce a simulation-based algorithm with competitive ratio $1 - \varepsilon^*(c_{\min})$ in Section 3.2 and a different simulation-based algorithm to guarantee a competitive ratio of at least $\frac{1}{2}$ (even for small $c_{\min}$) in Section 3.3. We finally present two simple hybrid algorithms that can obtain the best of two competitive ratios in Section 3.4.

3.1. High-Level Sketch of Our Approach

Let $\{y^*_{S,t,z_t}\}$ be the optimal assignment of Expected-LP $[\{F_t\}_{t=1}^T]$. As $\emptyset \in \mathcal{S}$, without loss of generality, we can only consider optimal assignments, where

$$\sum_{S \in \mathcal{S}} y^*_{S,t,z_t} = 1 \qquad \forall t \in [T],\ z_t \in \mathcal{Z}_t.$$

All of our simulation-based online algorithms in this paper follow four steps:

- At time $t = 0$ (before starting):
  (i) Preprocessing: Compute an optimal assignment $\{y^*_{S,t,z_t}\}$ of Expected-LP$[\{F_t\}_{t=1}^T]$ by invoking the offline oracle described in Assumption 3. Also, compute any other offline parameters that are occasionally needed by the algorithm.
- At each time $t = 1, 2, \ldots, T$:
  (ii) Simulation: Upon realizing consumer type z_t at time t, an outer procedure suggests $\hat{S} \in \mathcal{S}$ to

be assorted by sampling $\hat{S}$ from the distribution $\{y^*_{S,t,z_t}\}_{S\in\mathcal{S}}$ over $\mathcal{S}$.

(iii) Discarding: For each product $i \in \hat{S}$, a separate inner discarding procedure decides whether to remove this product from the final assortment given the history up to time t and realized type z_t. If no units of product i are available on hand, it is discarded automatically to guarantee inventory feasibility. Otherwise, the inner procedure of product i decides to discard or not. Let $\overline{S} \subseteq \hat{S}$ be the set of undiscarded products.

(iv) Postprocessing: Given z_t, $\hat{S}$, and $\overline{S}$, pick a probability distribution $\mathcal{F}_{z_t,\hat{S},\overline{S}}$ over all subsets of $\overline{S}$. Then, sample an assortment $\tilde{S} \sim \mathcal{F}_{z_t,\hat{S},\overline{S}}$ and offer it to the consumer.

In this four-step layout, Step (ii) is a loss-less randomized rounding for the optimal solution of Expected-LP $[\{F_t\}_{t=1}^T]$; however, the resulting assortment only guarantees inventory feasibility of each product in expectation. The role of Step (iii) and Step (iv) is to identify a (randomized) subset of this feasible in expectation assortment to not only guarantee inventory feasibility in each sample path but to also guarantee that the expected loss because of discarded products is small.

3.2. Large Initial Inventory: Toward Competitive Ratio $1-\varepsilon^*(c_{\min})$

The main idea behind the algorithm of this subsection is discarding each product independently at random with probability $\gamma = O(\sqrt{\log(c_{\min})/c_{\min}})$ in Step (iii) at each time t. Intuitively speaking, this discarding tries to leave enough probability for not violating any of the inventory constraints at each time t. To see this, if discarding a product does not change the choice probability of another assorted product, the expected number of unavailable units of each product i at each time t is at most $(1-\gamma)c_i$ because of the feasibility in expectation of sampled sets in Step (ii). Now consider the rental indicator random variables of product i, that is, random variables indicating whether this product is rented at each time or not. If these random variables are mutually independent across time, then we can use simple concentration bounds for the sum of independent random variables to prove our claim.

There are two major issues with this approach:

(i) Under weak substitutability (Assumption 1), discarding product i weakly increases the choice probability of another assorted product $j \neq i$. Therefore, the probability of an available unit of product j being rented at each time $\tau < t$ becomes larger than expected, which, in turn, increases the expected number of unavailable units of this product at time t if we only simulate the expected LP's optimal solution and discard each product independently with probability γ.

(ii) As resources are reusable and inventories are limited, the rental indicator random variable of product i at time $\tau < t$ is possibly positively correlated with the rental indicator random variable of the same product at time t; in fact, the first indicator forces the second indicator to be zero when the realized rental duration d_τ at time t is no smaller than $t-\tau$, the last unit of the product is rented at time τ, and no units of the product return during $[\tau+1, t]$.

We address the first issue in Section 3.2.1 by changing the algorithm and address the second issue in Section 3.2.2 by modifying the analysis.

3.2.1. Subassortment Sampling.

To fix the first issue, we propose the *subassortment sampling* procedure—a postprocessing procedure to be used in Step (iv). This procedure ensures that products that were not discarded in Step (ii) are rented by the arriving consumer with *exactly* the same probability as in the optimal solution of the expected LP benchmark. More formally, the subassortment sampling induces a distribution $\mathcal{F}_{z_t,\hat{S},\overline{S}}$ over subsets of $\overline{S}$ at each time t so that

$$\forall i \in \overline{S}: \qquad \mathbf{E}_{\tilde{S}\sim\mathcal{F}_{z_t,\hat{S},\overline{S}}}[\phi^{z_t}(\tilde{S}, i)] = \phi^{z_t}(\hat{S}, i). \tag{2}$$

It is not clear a priori whether such a distribution $\mathcal{F}_{z_t,\hat{S},\overline{S}}$ exists yet alone can be sampled from in polynomial time (polynomial in number of products n); nevertheless, for any general choice model satisfying weak substitutability (Assumption 1) and downward-closed feasibility (Assumption 2), we show such a distribution $\mathcal{F}_{z_t,\hat{S},\overline{S}}$ exists, and we introduce Procedure 1, which recursively samples a set from $\mathcal{F}_{z_t,\hat{S},\overline{S}}$ in polynomial time.[7]

Procedure 1 (Subassortment Sampling)

Input: choice model ϕ, assortment $S = \{1, 2, \ldots, m\}$, target probabilities $\{p_i\}_{i\in S}$

Let $\sigma : [m] \to [m]$ be a permutation such that $1 \geq \frac{p_{\sigma(1)}}{\phi(S,\sigma(1))} \geq \frac{p_{\sigma(2)}}{\phi(S,\sigma(2))} \geq \cdots \geq \frac{p_{\sigma(m)}}{\phi(S,\sigma(m))} \geq 0$

`/* Define` $\frac{p_{\sigma(j)}}{\phi(S,\sigma(j))} = 1$ `if` $\phi(S,\sigma(j)) = 0$ `or` $S = \emptyset$. `*/`

If $\frac{p_{\sigma(m)}}{\phi(S,\sigma(m))} = 1$ **then**

  return $\tilde{S} \leftarrow S$

else

  Let $q_0 = 1 - \frac{p_{\sigma(1)}}{\phi(S,\sigma(1))}$, $q_m = \frac{p_{\sigma(m)}}{\phi(S,\sigma(m))}$, and $q_j = \frac{p_{\sigma(j)}}{\phi(S,\sigma(j))} - \frac{p_{\sigma(j+1)}}{\phi(S,\sigma(j+1))}$ for $j = 1, \ldots, m-1$

  `/* Note that` $\sum_{j=0}^m q_j = 1$ `*/`

  Sample $j^* \sim \{q_j\}_{j=0}^m$

  If $j^* = 0$ **then**

    return $\emptyset$

  if $j^* = m$ **then**

    return S

  Let $S' \leftarrow \{\sigma^{-1}(j)\}_{j=1}^{j^*}$

  return $\tilde{S} \leftarrow$ SUB-ASSORTMENT SAMPLING$(\phi, S', \{\phi(S, i)\}_{i\in S'})$

Proposition 3. *For any weak substitutable and downward-closed feasible choice model ϕ, any assortment $S \in \mathcal{S}$, and any target probabilities $\{p_i\}_{i\in S}$ such that $p_i \le \phi(S,i)$ for all $i \in S$, Procedure 1 outputs a randomized assortment $\tilde{S}$ that satisfies (i) $\tilde{S} \subseteq S$, and (ii) $\mathbf{E}_{\tilde{S}}[\phi(\tilde{S},i)] = p_i$ for all $i \in S$. Moreover, it runs in time Poly(n).*

Remark 2. To guarantee Equation (2), given any $(z_t, \hat{S}, \overline{S})$ at Step (iv), we invoke Proposition 3 by setting $\phi \leftarrow \phi^{z_t}$, $S \leftarrow \overline{S}$, and $p_i \leftarrow \phi^{z_t}(\hat{S}, i)$ for all $i \in \overline{S}$. Note that $p_i = \phi^{z_t}(\hat{S},i) \le \phi^{z_t}(\overline{S},i)$ for all $i \in \overline{S}$, simply because of weak substitutability and the fact that $\overline{S} \subseteq \hat{S}$.

Proof of Proposition 3. Without loss of generality, we assume σ is the identity permutation, that is, $\sigma(i) = i$ for $i \in [m]$. To show the polynomial running time, observe that (a) the running time in each recursion is Poly(n), and (b) the number of iterations of this recursive algorithm is at most n because $|S| \le n$ at the beginning and the size of the S' that is the input of the next recursive call shrinks by one at each iteration; that is, $|S'| \le |S| - 1$.

Property (i) holds by construction. We show property (ii) by induction on $m = |S|$, that is, size of assortment S. In this induction, we use another simple property (iii) that $\phi(S,i)(\sum_{j=i}^{m} q_j) = p_i$ for all $i \in S$, which immediately hold by construction.

Base case ($m = 1$). In this case, Procedure 1 randomly outputs $\emptyset$ or S. By property (iii), the induction statement holds.

Inductive step ($m > 1$). Fix an arbitrary product $i \in S$. Notice that, by construction, $\mathbf{E}_{\tilde{S}}[\phi(\tilde{S},i)|j^* < i] = 0$, and $\mathbf{E}_{\tilde{S}}[\phi(\tilde{S},i)|j^* = m] = \phi(S,i)$. For any realized value, $j^* = i, \ldots, m-1$, and its corresponding $S' = \{1, \ldots, j^*\}$, we can use the induction hypothesis for the assortment S' with probabilities $p'_i = \phi(S,i)$ for each $i \in S'$. This is true simply because $|S'| \le m-1$ and that $\phi(S,i) \le \phi(S',i)$ for each $i \in S'$, as the choice model ϕ is weak substitute. By invoking the induction hypothesis when we use S' in the next recursive call, we have $\mathbf{E}_{\tilde{S}}[\phi(\tilde{S},i)|j^* = j] = \phi(S,i)$ for all $j = i, \ldots, m-1$. Thus, invoking property (iii),

$$\mathbf{E}_{\tilde{S}}[\phi(\tilde{S},i)] = \sum_{j=0}^{m} q_j \mathbf{E}_{\tilde{S}}[\phi(\tilde{S},i)|j^* = j] = \left(\sum_{j=i}^{m} q_j\right)\phi(S,i) = p_i,$$

which completes the inductive step and finishes the proof. □

3.2.2. The Algorithm and Analysis. Now, we present our first simulation-based algorithm (Algorithm 2) with its competitive ratio guarantee (Theorem 1). We defer the formal proof of Theorem 1 to Section EC5 in the Online Appendix.

Algorithm 2 (Simulation-Based Algorithm with Random Discarding)

Input: discarding probability $\gamma \in [0,1]$
- Preprocessing: Compute the optimal assignment $\{y^*_{S,t,z_t}\}$ of Expected-LP$[\{F_t\}_{t=1}^{T}]$ by invoking the offline assortment oracle (Assumption 3)
- **for** $t = 1$ *to* T **do**
  - `/* consumer` t `with type` $z_t \sim F_t$ `arrives */`
  - Simulation: Upon realizing consumer type z_t, sample $\hat{S}_t \sim \{y^*_{S,t,z_t}\}_{S\in\mathcal{S}}$
  - Discarding: Initialize $\overline{S}_t \leftarrow \hat{S}_t$
  - **for** *each product* $i \in \hat{S}_t$ **do**
    - Flip an independent coin and remove i from $\overline{S}_t$ with probability γ
    - **if** *there is no available unit of product* i **then**
      - Remove i from $\overline{S}_t$
  - Postprocessing: Let $\tilde{S}_t \leftarrow$ Sub-assortment Sampling $(\phi^{z_t}, \overline{S}_t, \{\phi(\hat{S}_t, i)\}_{i\in\overline{S}_t})$
  - `/* Send a query call to Procedure 1 with appropriate input arguments */`
  - Offer assortment $\tilde{S}_t$ to consumer t

Theorem 1. *By setting $\gamma = \gamma^*(c_{\min})$, the competitive ratio of Algorithm 2 against the Bayesian expected LP benchmark Expected-LP$[\{F_t\}_{t=1}^{T}]$ is at least $1 - \varepsilon^*(c_{\min}) = 1 - O(\sqrt{\log(c_{\min})/c_{\min}})$. Moreover, it runs in time Poly($n, T, \sum_{t\in[T]}|\mathcal{Z}_t|$) given oracle access to an offline algorithm for assortment optimization (Assumption 3).*

Before presenting the Proof Sketch of Theorem 1, we first discuss how to use a simple concentration argument to obtain a competitive ratio upper bound of $1 - O(\sqrt{\log(c_{\min} nT)/c_{\min}})$ for a slightly modified version of Algorithm 2. The simple concentration argument works as follows: consider a modified version of Algorithm 2 where the random discarding step is fully correlated. Namely, instead of flipping an independent coin for each product $i \in \overline{S}_t$, we flip a single coin and set $\overline{S}_t$ as all available products in $\hat{S}_t$ with probability γ, and empty set otherwise. Consider this modified Algorithm 2 until the first time that one of the sampled products is not available. Note that before such a bad event happens, all allocations are independent over consumers, and the subassortment sampling step is trivial (i.e., return $\tilde{S}_t = \overline{S}_t$ deterministically). We can bound the probability of this bad event at each time and for each product by an exponentially small probability in $c_{\min}$ because of the Chernoff bound. Applying union bound for every time $t \in [T]$ and every $i \in [n]$, and using the fact that when such a bad event happens, we have no control over the revenue, the resulting competitive ratio can be upper bounded by $1 - O(\sqrt{\log(c_{\min} nT)/c_{\min}})$. Note that this competitive ratio has an extra $\sqrt{\log(nT)}$ dependence and, thus, is strictly worse than the competitive ratio $1 - \varepsilon^*(c_{\min})$ stated in Theorem 1, which is independent of the number of products n and the number of consumers T.

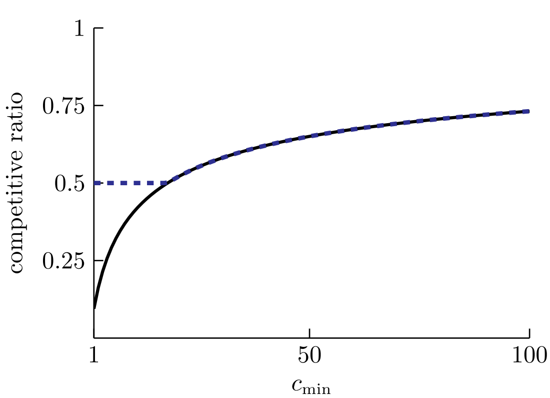
To prove Theorem 1 with the competitive ratio $1-\varepsilon^*(c_{\min})$ that is independent of the number of products n and the number of consumers T, we use a careful coupling argument in our analysis of Algorithm 2 that couples the rental indicator random variables of our algorithm with an alternative hypothetical algorithm. This hypothetical algorithm *ignores* inventory constraints of all the products and only simulates the expected LP's optimal solution combined with independent discarding of each product with probability γ. This algorithm generates an independent sequence of rental indicator random variables, allowing us to use simple concentration bounds. Importantly, this coupling trick is only possible because of the guarantee of the subassortment sampling procedure in Equation (2) (see the formal proof in Section EC5 in the Online Appendix).

Remark 3. Our analysis in this section mainly focuses on the asymptotic regime where $c_{\min}$ is large; however, we can still plot the competitive ratio $1-\varepsilon^*(c_{\min})$ of Algorithm 2 for small values of $c_{\min}$ by numerically evaluating $\varepsilon^*(c_{\min})$ using Equation (1). See the black solid curve in Figure 1.

3.3. Small Initial Inventory: Toward Competitive Ratio $\frac{1}{2}$

In this subsection, we propose our second simulation-based algorithm. The main difference between this algorithm and Algorithm 2 is in the discarding step: if no units of product i are available on hand, it is discarded automatically to guarantee inventory feasibility; otherwise, it is only selected in the final assortment if $r_i^{z_t} \geq P_{i,t}^{z_t}$, where $P_{i,t}^{z_t}$ are *nonadaptive* thresholds computed by the algorithm up front (this will be discussed later). The aim of this discarding procedure is to guarantee that only available products with high enough rental fees are assorted.

Figure 1. (Color online) Competitive Ratio of Simulation-Based Algorithms



Note. The black solid curve corresponds to Algorithm 2; the blue dashed curve corresponds to the hybrid between Algorithms 2 and 3 (see Section 3.4).

Technically speaking, for each product I, one can consider a separate DP to optimally make discarding decisions. This DP will maximize per-unit revenue to go of assorted units of product i over the finite time horizon $[1:T]$, given the randomized suggestion of Step (ii) (recall Section 3.1). The drawback is the need for a high-dimensional state variable that keeps track of the on-hand product inventories as well the inventory of units of the product that are in use (and will return to inventory at different times). A major ingredient of our algorithm is to replace this high-dimensional DP with a simple one that is *inventory independent* and uses an optimistic upper bound of c_i on the actual inventory in the Bellman equation for updating optimal per-unit revenue to go of product i.

3.3.1. Dynamic Programming for Per-Unit Revenue To Go with Replenishment.

In the rest of this subsection, let $X_{S,t,z_t} \triangleq y^*_{S,t,z_t} F_t(z_t)$ for every S,t and z_t. Suppose at each time t, a new independent consumer type $z_t \sim F_t$ is realized. Let $\hat{S} \sim X_{S,t,z_t}$ denote the randomized subset sampled in Step (ii) (simulation step). Fix a product $i \in [n]$ with initial inventory of c_i units. Now, consider a hypothetical scenario where an exogenous process *replenishes the inventory* at each time to guarantee we always have c_i units of the product on hand, no matter how many units are currently under rental. In this new problem, the goal is to design an online policy to discard or accept units of the reusable product once suggested in Step (ii) in order to maximize the per-unit revenue to go of renting this product. We can formulate this problem using a simple dynamic programming where $\mathcal{V}_{i,t}$ is the optimal per-unit revenue to go of product i during time interval $[t:T]$. Compared with the original high-dimensional DP for solving the optimum discarding, this DP is *optimistic* in that it "imagines" the deficiency in inventory is replenished every period.

As a convention, let $\mathcal{V}_{i,T+1}=0$. To write the Bellman update equation of the optimistic DP at time t using backward induction, suppose type z_t is realized and $\hat{S}=S$ (which happens w.p. X_{S,t,z_t}). If optimal policy decides to discard product i, then per-unit revenue to go will be $\mathcal{V}_{i,t+1}$. If optimal policy decides to not discard i, then, with probability $(1-\phi^{z_t}(S,i))$, the per-unit revenue to go will still be $\mathcal{V}_{i,t+1}$.[8] However, with probability $\phi^{z_t}(S,i)$, the consumer rents one of the c_i units (remember that inventory will always be full) and therefore generates a total revenue to go of $(c_i-1)\mathcal{V}_{i,t+1}$ (i.e., because of contribution of units not rented at time t; these units will transfer to the inventory at time $t+1$) plus $r_i^{z_t}+\mathcal{V}_{i,t+d}$ upon realization of rental time $d \sim G_i^{z_t}$ (i.e., because of the contribution of the rented unit). To summarize, we will have the following Bellman update equation:

$$\mathcal{V}_{i,t} = \sum_{z_t \in \mathcal{Z}_t} \sum_{S \in \mathcal{S}} X_{S,t,z_t} \times \max\Bigg\{ \mathcal{V}_{i,t+1},\ (1-\phi^{z_t}(S,i))\mathcal{V}_{i,t+1} + \phi^{z_t}(S,i)\left(\frac{1}{c_i}\sum_d g_i^{z_t}(d)(r_i^{z_t} + \mathcal{V}_{i,t+d}) + \frac{c_i - 1}{c_i}\mathcal{V}_{i,t+1}\right)\Bigg\}. \tag{3}$$

Note the update rule of the aforementioned dynamic programming can be simplified by rearranging the terms. Interestingly, the rule will be independent of c_i and $\phi^{z_t}(S,i)$, as they cancel out:

$$[\text{At time } t \text{ with type } z_t, i \in \hat{S} \text{ will be accepted}] \iff r_i^{z_t} \geq \mathcal{V}_{i,t+1} - \sum_d g_i^{z_t}(d)\mathcal{V}_{i,t+d}. \tag{4}$$

Remark 4. Later, in Sections EC3 and EC7 in the Online Appendix, we will replace this simple DP with a slightly modified one that has an *inventory-dependent* state but is still low-dimensional when rental times are infinite. This allows us to obtain (an almost) optimal competitive ratio for this special case.

3.3.2. The Algorithm and Analysis.

We now present our second simulation-based algorithm (Algorithm 3), with its competitive ratio guarantee (Theorem 2).[9]

Algorithm 3 (Simulation-Based Algorithm with Nonadaptive Per-Unit Revenue Thresholds)

<u>Preprocessing:</u>
- Compute the optimal assignment $\{y^*_{S,t,z_t}\}$ of Expected-LP$[\{F_t\}_{t=1}^T]$ by invoking the offline assortment oracle (Assumption 3)
- Set $X_{S,t,z_t} \triangleq y^*_{S,t,z_t} F_t(z_t)$ for every S,t and z_t where optimal assignment has a nonzero entry
- Solve the dynamic programming with Bellman update described in Equation (3) and boundary condition $\mathcal{V}_{i,T+1} = 0$ for every product i to obtain $\{\mathcal{V}_{i,t}\}_{i\in[n],t\in[T]}$
- Let $P_{i,t}^{z_t} \triangleq \mathcal{V}_{i,t+1} - \sum_d g_i^{z_t}(d)\mathcal{V}_{i,t+d}$, for all $i \in [n]$, $t \in [T]$, $z_t \in \mathcal{Z}_t$.

for $t = 1$ *to* T **do**
  `/* consumer` t `with type` $z_t \sim F_t$ `arrives */`
  <u>Simulation:</u> Upon realizing consumer type z_t, sample $\hat{S}_t \sim \{y^*_{S,t,z_t}\}_{S\in\mathcal{S}}$
  <u>Discarding:</u> Initialize $\overline{S}_t \leftarrow \hat{S}_t$
  for *each product* $i \in \hat{S}_t$ **do**
    if $r_i^{z_t} < P_{i,t}^{z_t}$ *or there is no available unit of product* i **then**
      Remove i from $\overline{S}_t$
  `/* Per-unit revenue thresholds` $\{P_{i,t}^{z_t}\}$ `are computed once, i.e., are nonadaptive */`
  <u>Postprocessing:</u> Let $\tilde{S}_t \leftarrow$ Sub-assortment Sampling $(\phi^{z_t}, \overline{S}_t, \{\phi(\hat{S}_t, i)\}_{i\in\overline{S}_t})$
  `/* Send a query call to Procedure 1 with appropriate input arguments */`
  Offer assortment $\tilde{S}_t$ to consumer t

Theorem 2. *The competitive ratio of Algorithm 3 against offline Bayesian expected LP benchmark, that is, Expected-LP $[\{F_t\}_{t=1}^T]$, is at least 1/2. Moreover, it runs in time Poly(n, T, $\sum_{t\in[T]}|\mathcal{Z}_t|$) given oracle access to an offline algorithm for assortment optimization (Assumption 3).*

Proof Sketch of Theorem 2. The running time is proved by Proposition 2 and the fact that the simple DP in Section 3.3.1 can be solved in polynomial time. The analysis of the competitive ratio can be decoupled across products. For each fixed product i, we do the analysis in two parts, each sketched as follows (see full details in Section EC6 in the Online Appendix):

- `Part (i)`, Section EC6.1: we first compare Algorithm 3 with the simple optimistic dynamic programming described in Section 3.3.1 and show the total expected revenue of Algorithm 3 because of rentals of product i is at least $c_i\mathcal{V}_{i,1}$. We prove this claim using induction and the fact that showing a subset $\tilde{S}_t$ of sampled assortment $\hat{S}_t$ can only increase the revenue to go of the discarding policy that follows the thresholds of the optimistic DP (as in the algorithm).
- `Part (ii)`, Section EC6.2: We then compare this simple dynamic programming with expected LP benchmark and show for each product i, $c_i\mathcal{V}_{i,1}$ is at least 1/2 of the contribution of product i to the optimal objective value of Expected-LP$[\{F_t\}_{t=1}^T]$ (part (ii)). In order to prove this part, we use the connection between the optimistic DP of Section 3.3.1 and a related factor–revealing LP that characterizes the competitive ratio of the optimistic DP. This connections leads us to apply duality arguments to find a lower bound on the ratio of $c_i\mathcal{V}_{i,1}$ and the contribution of product i to the optimal objective value of Expected-LP$[\{F_t\}_{t=1}^T]$.

3.4. Hybrid Between Algorithm 2 and Algorithm 3

3.4.1. Best of Both Worlds Discarding.

In both Algorithms 2 and 3, we have discarding policies (one for each of the products) that run independently from each other. Also, both competitive ratio analyses essentially decouple across different products, as we analyze the revenue performances of these discarding policies for each product separately. Moreover, in both analyses, we compare the expected revenue of each product i with the contribution of that product in the expected LP.

Considering all of these design and analysis aspects of our two algorithms, we can propose a hybrid algorithm where we decide on the choice of the discarding policy for each product i based on its initial inventory c_i upfront. Once we finalize these choices, we run the (possibly different) discarding algorithms in parallel and separately for different products during the discarding step of our final hybrid algorithm. In order to achieve the best of both worlds revenue performance guarantees of small and large inventory regimes, we partition the set of products into those with large initial inventory and those with small initial inventory (will be formally defined later) at the beginning. Given this partition, we assign a "randomized discarding" policy (as described in in Algorithm 2) to make discarding decisions of product i across times $t \in [T]$ if c_i is large, and we use a "discarding with per-unit revenue thresholds" (as described in in Algorithm 3) if c_i is small.

To distinguish between large and small c_i, we first solve the dynamic programming of the optimistic DP discarding policy in Section 3.3 for each product $i \in [n]$ by using its Bellman update equation (described in Equation (3)). We then use the value function $\mathcal{V}_{i,1}$ of the optimistic DP for product i to label this product as either large or small inventory. In particular, define $\mathcal{R}_i$ to be the ratio between the expected revenue to go of the optimistic DP for product i and the contribution of this product to the expected-LP's objective; that is,

$$\mathcal{R}_i \triangleq \frac{c_i \mathcal{V}_{i,1}}{\sum_{t=1}^{T} \sum_{z_t \in \mathcal{Z}_t} \sum_{S \in \mathcal{S}} X_{S,t,z_t} \phi^{z_t}(S,i) r_i^{z_t}}.$$

Note that $\mathcal{R}_i \in [0.5, 1]$ from Theorem 2. Also note that $\varepsilon^*(c)$ is continuous and monotone increasing in c, $\varepsilon^*(0) = 1$, and $\varepsilon^*(+\infty) = 0$—see Equation (1) for the definition of $\varepsilon^*(\cdot)$. We next compare the ratio $\mathcal{R}_i$ with $1 - \varepsilon^*(c_i)$. In fact, if $\mathcal{R}_i + \varepsilon^*(c_i) < 1$, we then except the competitive ratio of randomized discarding to be no smaller than that of the optimistic DP, and hence, we label the product as "large inventory." Otherwise, we expect the optimistic DP to beat the randomized discarding in terms of the competitive ratio, and hence, we label the product as "small inventory." Finally, we perform the subassortment sampling as a postprocessing step in order to correct the choice probabilities of nondiscarded products (which might have been increased because of weak substitution and that other products are either discarded or not even selected in the assortment as they were not available at the first place). We denote the resulting algorithm by `Sim+Hybrid(i)`.

Theorem 3. *The competitive ratio of* `Sim+Hybrid(i)` *against offline Bayesian expected LP benchmark, that is, Expected-LP*$[\{F_t\}_{t=1}^T]$*, is at least* $1 - \min\left(\frac{1}{2}, \varepsilon^*(c_{\min})\right)$ *(see Figure 1).*

Proof Sketch. As it can be seen by following the lines of both the proof of Theorem 1 in Section 3.2.2 and the Proof of Theorem 2 in Section EC6 in the Online Appendix, the arguments for the revenue performance of the corresponding discarding policies for each product i (and also the resulting competitive ratio) are independent of how discarding of another product i' is handled, and therefore, the competitive ratio guarantees of the randomized discarding and the optimistic DP from these theorems still hold for the hybrid algorithm. For brevity, we do not repeat these proofs. We conclude that the proposed hybrid algorithm has a competitive ratio of at least $\max\left(\frac{1}{2}, 1 - \varepsilon^*(c_{\min})\right)$. □

3.4.2. Monte Carlo Simulation to Help. In principle, given the sequence of type distributions $\{F_t\}_{t=1}^T$, one can simulate both Algorithms 2 and 3 and estimate their expected future revenues using Monte Carlo simulation, starting at any time $t \in [T]$ (given any history up to time t). Now, a simple hybrid algorithm can switch to the algorithm with the higher expected revenue and runs this algorithm for the next time step given the current history of rental products. By repeatedly applying this method at each time t given the history up to this time—a techniques known as the *method of conditional expectation*—we end up with an alternative hybrid algorithm that essentially is the be-the-leader policy among the two policies at each time, meaning that its expected future revenue at each time is at least the expected future revenue of each of the two policies (which can be proved using induction). Hence, this hybrid algorithm clearly obtains the best of both worlds competitive ratio of $1 - \min\left(\frac{1}{2}, \varepsilon^*(c_{\min})\right)$, similar to our previous hybrid algorithm. Such a policy can also choose to switch at a lower frequency, but no matter what frequency it picks, it is expected to outperform both policies in expectation, both in theory and practice. We use this alternative hybrid algorithm, which is denoted by `Sim+Hybrid(ii)`, in our numerical simulations in Section EC8 in the Online Appendix as well.

Theorem 4. *The competitive ratio of* `Sim+Hybrid(ii)` *against offline Bayesian expected LP benchmark, that is, Expected-LP*$[\{F_t\}_{t=1}^T]$*, is at least* $1 - \min\left(\frac{1}{2}, \varepsilon^*(c_{\min})\right)$ *(see Figure 1).*

Proof Sketch. Consider the following backward induction. The induction hypothesis is that given current state $\mathbf{J}$ (i.e., the number of available units of each product in the inventory, as well as the return time of each allocated unit of every product), and time period t, the expected revenue from time t to time T in `Hybrid-Sim(ii)` is weakly higher than Algorithms 2 and 3. Base case $t = T$ is straightforward. Suppose the induction hypothesis is correct for $t+1, \ldots, T$. Consider the inductive step for time t and current state $\mathbf{J}$. Suppose Monte Carlo simulation suggests that given state $\mathbf{J}$, Algorithm 2 achieves higher expected revenue from time t to time T, and its induced new state is $\mathbf{J}'$ (the analysis for the other case is similar). In this case, we know that the expected revenue in `Hybrid-Sim(ii)` from time t to time T can be decomposed into the following two terms: (i) the expected revenue in `Hybrid-Sim(ii)` at time t, and (ii) the expected revenue in `Hybrid-Sim(ii)` from time $t+1$ to time T under state $\mathbf{J}'$. By construction, term (i) equals to the expected revenue in Algorithm 2 at time t. By our induction hypothesis for time $t+1$ with state $\mathbf{J}'$, term (ii) is weakly higher than the expected revenue in Algorithm 2 from time $t+1$ to time T given state $\mathbf{J}'$. Hence, the expected revenue in `Hybrid-Sim(ii)` from time t to time T given state $\mathbf{J}$ is weakly higher than Algorithms 2 and 3 as well, which concludes the backward induction. □

Remark 5. We would like to highlight that whereas both `Sim+Hybrid(i)` and `Sim+Hybrid(ii)` attain the theoretical best of both worlds competitive ratio that was mentioned earlier, and as we see in our numerical simulations in Section EC8 in the Online Appendix, they both outperform other existing policies in practical scenarios of our problem; they differ in terms of computational requirements. In fact, `Sim+Hybrid(i)` can easily make upfront decisions for the choice of discarding policy of each product i with almost no extra computation compared with Algorithms 2 and 3; nevertheless, `Sim+Hybrid(ii)` needs to run Monte Carlo simulations several times (depending on the switching frequency) by sampling from future types, which makes it less practically appealing.

4. Conclusion

We studied designing near-optimal algorithms for the online assortment of reusable resources in the Bayesian setting. We proposed an algorithmic framework based on four modular steps: (i) solving the expected LP, (ii) simulating the solution, (iii) running a separate discarding procedure for each product to maintain point-wise inventory feasibility (while only losing a negligible fraction of the revenue of each product), and (iv) performing a postprocessing step to adjust choice probabilities of nondiscarded items. Using this framework, we designed an algorithm that is $1-\min\left(\frac{1}{2}, O\left(\sqrt{\log(c_{\min})/c_{\min}}\right)\right)$ under the general rental duration distributions and an improved near-optimal algorithm with competitive ratio $1-1/\sqrt{(c_{\min}+3)}$ under infinite rental durations. Not only do our algorithms outperform the existing algorithms in the literature theoretically, we further verified their revenue performance advantages through numerical simulations.

As a roadmap for future, it is interesting to study what other practical aspects of a real-world assortment problem beyond reusable resources can be modeled and to what extent mathematical programming techniques can be used to design competitive algorithms there. On the technical side, the most immediate open problem stemming from our work is finding the optimal competitive ratio for the case of general rental duration distributions. In particular, can one shave the logarithmic factor in our competitive ratio and obtain a $1-O(1/\sqrt{c_{\min}})$ competitive algorithm, similar to the best known competitive ratio in the nonreusable case? As a different yet more ambitious future direction, it would be interesting to study classes of stochastic online optimization similar to the Bayesian online assortment further in order to discover the computational hardness of computing or approximating the optimum online policy, that is, the DP policy. An interesting discovery here would be obtaining improved approximations against the optimum online benchmark versus the expected LP benchmark through polynomial time policies, such as Anari et al. (2019), or proving its impossibility.

Acknowledgments

The authors thank the anonymous associate editor and the referees for their comments and feedback throughout the revision process. R. Niazadeh's research is partially supported by an Asness Junior Faculty Fellowship from the University of Chicago Booth School of Business.

Endnotes

[1] For example, see Papadimitriou and Tsitsiklis (1987) for PSPACE-hardness of finding the optimum policy in partially observable Markov decision processes, and see related discussions in Anari et al. (2019) and Rusmevichientong et al. (2020).

[2] If rental fees are type dependent, Rusmevichientong et al. (2020) show the same policy obtains $1-\min\left(\frac{1}{2}, \frac{R}{\sqrt[3]{c_{\min}}}\right)$ competitive ratio guarantee, where R is the ratio between maximum and minimum rentals fees across different types.

[3] See specifically the primal routing algorithm in section 7 of Gallego et al. (2015) and the separation algorithm in section 4.2 of Wang et al. (2018).

[4] We would like to highlight that after appearance of an online version of our paper, through a personal communication with authors of Goyal et al. (2020), we were informed that this paper (which was not available online at the time) independently and concurrently discovered a procedure similar to our subassortment sampling for settings with adversarial arrival and reusable resources.

[5] It is worth noting that the $\frac{1}{2}$-competitive approximate DP algorithm of Rusmevichientong et al. (2020) cannot be combined using this approach with our near-optimal discarding policy, as this approximate DP algorithm competes with the optimum online policy and not the expected LP.

[6] In fact, one needs to run the ellipsoid method for the dual of this LP using the offline assortment solver as the separation oracle in order to find the optimal solution. In practice, to obtain a faster algorithm, one can use *cutting plane* methods such as Vaidya (1996) or even faster almost-linear-time cutting plane methods such as Lee et al. (2015) that use the separation oracle more efficiently.

[7] It is noteworthy that Goyal et al. (2020) independently and concurrently discovered an idea similar to our subassortment sampling for settings with adversarial arrival and reusable resources.

[8] In this hypothetical scenario, we assume that the probability that the consumer select product i equals $\phi^{z_t}(S,i)$ regardless of whether another product i' is discarded from S.

[9] The competitive ratio in Theorem 2 is optimal even if rental times are infinite. Consider the following example: there is a single nonreusable product with a single unit. There are two time periods, $T=2$.

Consumer 1 has a deterministic type that deterministically purchases this item with fee 1. With probability ϵ, consumer 2 has a type that deterministically purchases this item with fee $1/\epsilon$. Otherwise (i.e., with probability $1-\epsilon$), consumer 2 has a type that purchases nothing. In this example, the expected revenue of the Bayesian expected LP benchmark, as well as the clairvoyant policy, is $2-\epsilon$, whereas the expected revenue of any online policy is, at most, one.

References

Adelman D (2007) Dynamic bid prices in revenue management. *Oper. Res.* 55(4):647–661.

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

Alaei S, Hajiaghayi MT, Liaghat V (2012) Online prophet-inequality matching with applications to ad allocation. *Proc. 13th ACM Conf. Electr. Commerce* (Association for Computing Machinery, New York), 18–35.

Anari N, Niazadeh R, Saberi A, Shameli A (2019) Nearly optimal pricing algorithms for production constrained and laminar Bayesian selection. *Proc. 2019 ACM Conf. Econom. Comput.* (Association for Computing Machinery, New York), 91–92.

Baek J, Ma W (2022) Bifurcating constraints to improve approximation ratios for network revenue management with reusable resources. *Oper. Res.* 70(4):2226–2236.

Chawla S, Hartline JD, Malec DL, Sivan B (2010) Multi-parameter mechanism design and sequential posted pricing. *Proc. 42nd ACM Sympos. Theory Comput.* (Association for Computing Machinery, New York), 311–320.

Devanur NR, Sivan B, Azar Y (2012) Asymptotically optimal algorithm for stochastic adwords. *Proc. 13th ACM Conf. Electr. Commerce* (Association for Computing Machinery, New York), 388–404.

Dickerson J, Sankararaman K, Srinivasan A, Xu P (2018) Allocation problems in ride-sharing platforms: Online matching with offline reusable resources. *Proc. AAAI Conf. Artificial Intelligence*, vol. 32 (AAAI Press, Washington, DC).

Gallego G, Li A, Truong V-A, Wang X (2015) Online personalized resource allocation with customer choice. Preprint, submitted November 5, https://arxiv.org/abs/1511.01837.

Golrezaei N, Nazerzadeh H, Rusmevichientong P (2014) Real-time optimization of personalized assortments. *Management Sci.* 60(6): 1532–1551.

Goyal V, Iyengar G, Udwani R (2020) Online allocation of reusable resources via algorithms guided by fluid approximations. Preprint, submitted October 8, https://arxiv.org/abs/2010.03983.

Hajiaghayi MT, Kleinberg R, Sandholm T (2007) Automated online mechanism design and prophet inequalities. *AAAI'07 Proc. 22nd National Conf. Artificial Intelligence* (AAAI Press, Palo Alto, CA), 58–65.

Kök AG, Fisher ML, Vaidyanathan R (2008) Assortment planning: Review of literature and industry practice. Agrawal N, Smith S, eds. *Retail Supply Chain Management*, International Series in Operations Research & Management Science, vol. 223 (Springer, Boston), 99–153.

Lee YT Sidford A, Wong SC (2015) A faster cutting plane method and its implications for combinatorial and convex optimization. *2015 IEEE 56th Annual Sympos. Foundations Comput. Sci.* (IEEE, Piscataway, NJ), 1049–1065.

Ma W, Simchi-Levi D, Zhao J (2020) Dynamic pricing (and assortment) under a static calendar. *Management Sci.* 67(4):2292–2313.

Papadimitriou CH, Tsitsiklis JN (1987) The complexity of Markov decision processes. *Math. Oper. Res.* 12(3):441–450.

Rusmevichientong P, Sumida M, Topaloglu H (2020) Dynamic assortment optimization for reusable products with random usage durations. *Management Sci.* 66(7):2820–2844.

Rusmevichientong P, Shmoys D, Tong C, Topaloglu H (2014) Assortment optimization under the multinomial logit model with random choice parameters. *Prod. Oper. Management* 23(11): 2023–2039.

Vaidya PM (1996) A new algorithm for minimizing convex functions over convex sets. *Math. Programming* 73(3):291–341.

Wang X, Truong V-H, Bank D (2018) Online advance admission scheduling for services with customer preferences. Preprint, submitted May 26, https://arxiv.org/abs/1805.10412.

Zhang D, Adelman D (2009) An approximate dynamic programming approach to network revenue management with customer choice. *Transportation Sci.* 43(3):381–394.

Yiding Feng is a principal researcher at the University of Chicago Booth School of Business postdoctoral program. Before that, he was a researcher at Microsoft Research Laboratory New England (MSR). He earned his PhD in computer science with a focus on CS+ economics research in mechanism design, online algorithms, and information design at Northwestern Engineering. Before that, he received his BS degree from the ACM Honors Class at Shanghai Jiao Tong University.

Rad Niazadeh is an assistant professor of operations management and Asness Junior Faculty Fellow at the University of Chicago Booth School of Business. He primarily studies the theory and practice of data-driven sequential decision making and algorithmic mechanism design, with the goal of advancing the theoretical methodologies and practical foundations of online platforms, electronic marketplaces, and nonprofit/governmental agencies in dynamic and complex environments. He has received several awards for his research, including the Asness Junior Faculty Fellowship, INFORMS Auctions and Market Design 2021 Rothkopf Junior Researcher Paper Award (first place), the INFORMS Revenue Management and Pricing Dissertation Award (honorable mention), and the Google PhD Fellowship in Market Algorithms.

Amin Saberi is Professor of Management Science and Engineering at Stanford University. He received his BSc from Sharif University of Technology and his PhD from Georgia Institute of Technology in computer science. His research interests include algorithms, design and analysis of social networks, and applications. He is a recipient of the Terman Fellowship, Alfred Sloan Fellowship, and a number of best paper awards. Amin was the founding CEO and chairman of NovoEd Inc., a social learning environment designed in his research laboratory and used by universities such as Stanford, UC Berkeley, and University of Michigan as well as nonprofit and for-profit institutions for offering courses to hundreds of thousands of learners around the world.