An Approximation Algorithm for Network Revenue Management under Nonstationary Arrivals

Yuhang Ma[1], Paat Rusmevichientong[2], Mika Sumida[1], Huseyin Topaloglu[1]
[1] School of Operations Research and Information Engineering, Cornell Tech, New York, NY 10044,
[2] Marshall School of Business, University of Southern California, Los Angeles, CA 90089,
ym367@cornell.edu, rusmevic@marshall.usc.edu, ms3268@cornell.edu, topaloglu@orie.cornell.edu

August 16, 2019

We provide an approximation algorithm for network revenue management problems. In our approximation algorithm, we construct an approximate policy using value function approximations that are expressed as linear combinations of basis functions. We use a backward recursion to compute the coefficients of the basis functions in the linear combinations. If each product uses at most L resources, then the total expected revenue obtained by our approximate policy is at least $1/(1+L)$ of the optimal total expected revenue. In many network revenue management settings, although the number of resources and products can become large, the number of resources used by a product remains bounded. In this case, our approximate policy provides a constant-factor performance guarantee. Our approximate policy can handle nonstationarities in the customer arrival process. To our knowledge, our approximate policy is the first approximation algorithm for network revenue management problems under nonstationary arrivals. Our approach can incorporate the customer choice behavior among the products, and allows the products to use multiple units of a resource, while still maintaining the performance guarantee. In our computational experiments, we demonstrate that our approximate policy performs quite well, providing total expected revenues that are substantially better than its theoretical performance guarantee.

Key words: network revenue management, approximate dynamic programming, dynamic pricing

1. Introduction

In network revenue management problems, we manage the limited capacities for a collection of resources to satisfy the requests for different products that arrive randomly over time. Such problems find applications in a variety of settings, including airlines, hospitality, railways, and cloud computing. In airlines, for example, the resources are the flight legs and the products are the itineraries offered to customers that can consume capacities on multiple flight legs. In hospitality, on the other hand, the resources are the availabilities of hotel rooms on each day and the products are the multiple night stays offered to customers that can consume capacities on multiple days. The main tradeoff in network revenue management problems involves keeping a balance between accepting a product request that is currently in the system to generate some immediate revenue and reserving the resource capacities for a potentially more profitable product request that can arrive in the future. Nevertheless, the key difficulty in finding the optimal course of action arises from the fact that serving a request for a product consumes the capacities of the different resources used by the product. Thus, computing the optimal course of action requires keeping track of the remaining capacities for all the resources simultaneously, creating the curse of dimensionality as the number of resources increases.

In this paper, we provide an approximation algorithm for network revenue management problems. Our problem setup follows the standard network revenue management literature. We have resources with limited capacities that can be used to serve the requests for products arriving randomly over a finite selling horizon. At each time period in the selling horizon, a customer arrives with a request for a particular product. If we accept the product request, then we generate a certain revenue and consume the capacities of the different resources used by the product. The arrivals of the product requests can have arbitrary nonstationarities, but they are independent between the time periods. The goal is to find a policy to determine which product requests to accept to maximize the total expected revenue over the selling horizon. The dynamic programming formulation for this problem requires a high-dimensional state variable that keeps track of the remaining resource capacities. Thus, it is intractable to compute the optimal policy.

Contributions: Letting L be the maximum number of resources used by a product, we give an approximate policy that is guaranteed to obtain at least $1/(1+L)$ of the optimal total expected revenue. In many network revenue management settings, the number of resources and products can become large, but the number of resources used by a product remains bounded. In airlines, for example, L corresponds to the maximum number of flight legs included in an itinerary, which usually does not exceed two or three. When the number of resources used by a product is bounded, our approximate policy provides a constant-factor performance guarantee. To our knowledge, our approximate policy is the first approximation algorithm that can handle arbitrary nonstationarities in the arrivals of the product request. Also, we note that our performance guarantee is independent of the numbers of resources and products, and it does not involve any hidden constants that can potentially depend on other input data.

The idea behind our approximate policy is to use value function approximations that are expressed as linear combinations of basis functions. The coefficients in the linear combinations are computed through a backward recursion over the time periods in the selling horizon. The approach that we use to construct our approximate policy provides flexibility on two important dimensions. First, our value function approximations are a member of a relatively broad class. In our value function approximations, we have one basis function for each product. The basis function associated with each product takes the value zero when we do not have sufficient capacities to serve a request for the product. Therefore, we refer to our basis functions as availability-tracking basis functions. For any choice of availability-tracking basis functions, we can use our backward recursion over the time periods to compute coefficients for the basis functions in the linear combinations. In our backward recursion, we have a tuning parameter θ whose specific allowable values are determined by the availability-tracking basis functions that we use. We prove that if we construct an approximate policy using the value functions computed through our backward recursion, then the approximate policy is guaranteed to obtain at least $1/(1+\theta L)$ of the optimal total expected revenue. This result holds for any choice of availability-tracking basis functions. The performance guarantee of $1/(1+\theta L)$ improves as θ gets smaller. In our approach, the tuning parameter θ must be at least one, and there exist availability-tracking basis functions that permit choosing the smallest possible value of one for the tuning parameter θ, in which case, we obtain the performance guarantee of $1/(1+L)$ discussed in the previous paragraph.

Second, we start with a network revenue management setup where each customer arrives into the system to purchase a particular product and each product uses at most one unit of a resource. This setup allows us to convey the key ideas without notational clutter, but we can extend our approach to more general setups. In particular, we show that we can extend our approach to a case in which we offer a set of products to each customer, and the customer chooses among the offered products. Similarly, we show that we can extend our approach to a case in which a product may use more than one unit of a resource, which occurs, for example, in airlines when group reservations are allowed. If the customers choose among the offered products, then the performance guarantee of our approximate policy is still $1/(1+L)$, whereas if a product uses at most M units of a particular resource, then the performance guarantee of our approximate policy is $1/(1+(2M-1)L)$. Lastly, our backward recursion is simple and does not require solving any involved optimization problems, but it can use the solution to a linear programming approximation to construct our value function approximations, while retaining the performance guarantee of $1/(1+L)$.

Literature Review: One approach to construct policies in network revenue management problems is based on bid prices, where we associate a bid price for each resource, measuring the value of a unit of resource. In this case, we accept the request for a product if the revenue from the product exceeds the total value of the resources consumed by the product. Simpson (1989) and Williamson (1992) compute bid prices using a linear programming approximation that is constructed under the assumption that the numbers of product requests take on their expected values. They use the optimal values of the dual variables associated with certain capacity constraints to measure the value of a unit of resource. Talluri and van Ryzin (1998) show that such a bid price policy is asymptotically optimal, as the expected numbers of product requests and the capacities of the resources increase linearly with the same rate. Talluri and van Ryzin (1999) use samples of the product requests in the linear program to capture information about the distributions of the product requests. Bertsimas and Popescu (2003) measure the value of a unit of resource directly by perturbing the right side of a capacity constraint and resolving the linear program. A number of papers characterize the loss in the optimal total expected revenue for the policies derived from linear programming approximations; see Cooper (2002), Maglaras and Meissner (2006), Jasin and Kumar (2012), and Jasin and Kumar (2013). These papers focus on an asymptotic regime, where the total expected demand and the capacities of the resources increase linearly with the same rate. The characterization of the losses are additive and it depends on the input data. In contrast, our performance guarantee holds without an asymptotic regime. Also, our performance guarantee of $1/(1+L)$ is multiplicative and does not depend on the input data other than L.

The value of a unit of resource should depend on the time left in the selling horizon to utilize the resource, as well as the remaining capacity of the resource. Thus, bid prices should, in principle, be time and capacity dependent. There is work on computing such bid prices. Adelman (2007), Zhang and Adelman (2009), Kunnumkal and Topaloglu (2010a), and Kirshner and Nediak (2015) develop methods that yield time dependent bid prices, whereas Cooper and de Mello (2007), Topaloglu (2009), and Zhang (2011) develop methods that yield capacity dependent bid prices. Tong and Topaloglu (2013), Vossen and Zhang (2015a,b), and Kunnumkal and Talluri (2016a) show that some of these approaches are equivalent, although their derivations use seemingly unrelated paths. There is also work on incorporating customer choice behavior into network revenue management problems, where customers choose among the offered products. Some of the work extends and analyzes the linear programming approximation to incorporate customer choice behavior; see Gallego et al. (2004), Liu and van Ryzin (2008), Kunnumkal and Topaloglu (2008), Bront et al. (2009), Mendez-Diaz et al. (2010), Meissner et al. (2012), Talluri (2014), and Strauss and Talluri (2017). There is also work on approximating the value functions under customer choice behavior; see Zhang and Cooper (2005, 2009), Zhang and Adelman (2009), Kunnumkal and Topaloglu (2010b), and Kunnumkal and Talluri (2016b). There are papers that use stochastic approximation to compute booking limits and bid prices; see van Ryzin and Vulcano (2008a,b), Topaloglu (2008), and Chaneton and Vulcano (2011). These papers do not give performance guarantees.

Online packing problems are closely related to network revenue management problems, as each packing constraint may capture the capacity of a resource and each arriving item may capture the request for a product. Working with the random-order arrival model for the product requests, Kesselheim et al. (2014) give a policy with a competitive ratio of $1-O(\sqrt{(\log L)/c_{\min}})$, where $c_{\min}$ is the smallest resource capacity. The authors also give an upper bound of $1-\Omega(1-1/L^{\frac{1}{c_{\min}-1}})$ on the competitive ratio. Thus, their competitive ratio approaches one as the capacities of the resources become large. The strong competitive ratio in Kesselheim et al. (2014) comes at the cost of having to work with the random-order arrival model, which is restrictive for the network revenue management setting. In the random-order arrival model, the set of product requests that are to arrive over the selling horizon is fixed a priori, possibly by an adversary. Once the set of product requests that are to arrive is fixed, these product requests arrive according to a random permutation. A competitive ratio under the random-order arrival model implies the same competitive ratio under independent and identically distributed arrivals. However, the competitive ratio in Kesselheim et al. (2014) does not hold under nonstationary arrivals. Indeed, in Appendix E, we give a simple example to show that the policy proposed by Kesselheim et al. (2014) performs arbitrarily poorly under nonstationary arrivals. Online packing problems are often motivated by the adwords setting, where assuming stationary arrivals is reasonable, because the users doing a search today may not be too different from those doing a search tomorrow. However, assuming stationary arrivals is not reasonable in the network revenue management setting, because the customers booking airline tickets close to the departure time are clearly different from those booking early. We also tested the policy in Kesselheim et al. (2014) in our computational experiments under both stationary and nonstationary arrivals, though it does not have a competitive ratio in the latter case. The policy in Kesselheim et al. (2014) noticeably lags behind our approximate policy in both cases. Much of the other work on online packing problems also uses the random-order arrival model; see Devanur and Hayes (2009), Molinaro and Ravi (2014), Agrawal et al. (2014). Devanur et al. (2011) work with stationary or adversarial arrivals, but with restrictions on choices of the adversary. Tan and Srikant (2012) use nonstationary arrivals, but allow violations of capacity constraints.

The adwords problem is a special case of the online packing problem, where the revenue and resource consumption of a product request are both dictated by the bid that an advertiser places for a keyword. A number of papers use the primal-dual framework to design algorithms for the adwords problem; see Mehta et al. (2007), Buchbinder and Naor (2009), and Goel et al. (2010). In the primal-dual framework, a tradeoff function is used to prioritize the bidders for receiving keywords. For example, the tradeoff function may prioritize the bidders with the larger remaining budgets for receiving keywords. After making decisions over the selling horizon by following the tradeoff function, one uses the decisions to construct a primal-dual solution pair for a linear program that provides an upper bound on the optimal total expected revenue, in which case, studying the optimality gap of this solution yields the competitive ratio.

Motivated by product assortment personalization, Golrezaei et al. (2014) study a problem involving multiple products with limited inventories. The firm offers a set of products to an arriving customer. The customer makes a choice among the offered products according to a choice model. The goal is to find a policy to determine which set of products to offer to each customer to maximize the total expected revenue over the selling horizon. If each product uses exactly one resource so that $L = 1$, then our network revenue management problem under the customer choice behavior corresponds to the problem studied by Golrezaei et al. (2014). The authors use the primal-dual framework to construct a policy with a 50% competitive ratio. The tradeoff function used by Golrezaei et al. (2014) adjusts the revenue of each product as a function of its remaining inventory, prioritizing the products with larger inventories. Their policy offers a set of products to maximize the immediate expected adjusted-revenue from each customer. Our availability-tracking basis functions are somewhat analogous to the tradeoff functions in the primal-dual framework in the sense that our basis functions also adjust the revenue of each product as a function of the remaining inventories of the resources, in which case, we accept a product request only if the immediate adjusted-revenue from the product is non-negative. However, our basis functions, by themselves, are not sufficient to obtain a performance guarantee. To obtain our performance guarantee, we need to weigh our basis functions with coefficients and the computation of the coefficients requires designing a backward recursion over the time periods.

Recently, Rusmevichientong et al. (2017) consider dynamic assortment problems with random usage durations. In their problem setting, each customer uses a product for a random duration of time, before returning it. The authors use linear value function approximations to obtain a policy with a 50% performance guarantee. The slopes of the linear value function approximations are computed by using a backward recursion that resembles ours on the surface, but both papers tackle rather different challenges. In Rusmevichientong et al. (2017), there is no resource network. If a customer chooses a product, then she consumes only the inventory of this product. In the absence of a resource network, linear value function approximations are sufficient to get a performance guarantee, but under a resource network, linear approximations were not sufficient for us. When we use nonlinear basis functions, the opportunity cost of the resource capacities used by a product becomes dependent on the state of the system. Furthermore, since different products may use the same resource, it is difficult to account for the opportunity cost of each resource separately. We get around these difficulties by constructing a state-independent upper bound on the opportunity cost and constructing a backward recursion over the time periods that accounts for the optimal total expected revenue from a particular product. Nevertheless, we cannot claim that our work is a generalization of Rusmevichientong et al. (2017) either, because incorporating random usage durations in the presence of a resource network brings its own complications.

There is work in dynamic optimization with implications on revenue management. Chan and Farias (2009) give approximation algorithms for a certain class of stochastic control problems. Their work implies that if each product uses only one resource, then a myopic policy provides a 50% performance guarantee even under customer choice behavior. Asadpour and Nazerzadeh (2016) give algorithms for maximizing random monotone submodular functions. Their non-adaptive algorithm implies that if there is a single resource, then a static algorithm that a priori chooses the product requests to accept provides a performance guarantee of $(e-1)/(2e)$. Lastly, Wang et al. (2016) and Gallego et al. (2016) give algorithms with performance guarantees for dynamic resource allocation problems, but they also consider the case where each product uses one resource.

Organization: In Section 2, we give a dynamic programming formulation for the network revenue management problem. In Section 3, we construct our approximate policy, provide a performance guarantee, and show that this guarantee is tight. In Section 4, we give extensions of our approximate policy to the cases where the customers choose among the products and a product may consume more than one unit of a resource. We also discuss how to leverage a linear programming approximation when constructing our approximate policies. In Section 5, we provide computational experiments. In Section 6, we conclude.

2. Problem Formulation

The set of resources is $\mathcal{L}$ and the set of products is $\mathcal{J}$. The capacity of resource i is C_i. If we accept a request for product j, then we consume one unit of capacity of each resource in the set $A^j \subseteq \mathcal{L}$. We use L to denote the maximum number of resources that can be used by a product, so $L = \max_{j \in \mathcal{J}} |A^j|$. Accepting a request for product j generates a revenue of r_j. We have T time periods in the selling horizon indexed by $\mathcal{T} = \{1, \dots, T\}$. Each time period corresponds to a sufficiently small interval of time so that there is at most one product request at each time period. We get a request for product j at time period t with probability λ_j^t. With the remaining probability $1 - \sum_{j \in \mathcal{J}} \lambda_j^t$, there is no request for a product at time period t.

Our goal is to find a policy to determine which product requests to accept at each time period to maximize the total expected revenue over the selling horizon, while adhering to the capacity availabilities of the resources. To capture the state of the system, we let x_i be the remaining capacity of resource i at the beginning of a generic time period. Therefore, we can use the vector $\boldsymbol{x} = (x_i : i \in \mathcal{L}) \in \mathbb{Z}_+^{|\mathcal{L}|}$ to capture the state of the resources. The set of possible states is $\mathcal{Q} = \{\boldsymbol{x} \in \mathbb{Z}_+^{|\mathcal{L}|} : x_i \le C_i \ \forall\, i \in \mathcal{L}\}$. We can accept a request for product j when we have at least one unit of capacity for each resource used by product j. Therefore, letting $\mathbb{1}_{\{\cdot\}}$ be the indicator function, we can accept a request for product j if and only if $\prod_{i \in A^j} \mathbb{1}_{\{x_i \ge 1\}} = 1$.

We use $V^t(\boldsymbol{x})$ to denote the maximum total expected revenue over time periods $t, t+1, \dots, T$, given that the system is in state $\boldsymbol{x}$ at time period t. Letting $\boldsymbol{e}_i \in \mathbb{Z}_+^{|\mathcal{L}|}$ be the unit vector with a one in the i-th component and defining $[a]^+ = \max\{a, 0\}$, we can find the optimal policy by computing the optimal value functions $\{V^t(\boldsymbol{x}) : \boldsymbol{x} \in \mathcal{Q},\ t \in \mathcal{T}\}$ through the dynamic program

$$\begin{aligned} V^t(\boldsymbol{x}) \;&=\; \sum_{j\in\mathcal{J}} \lambda_j^t \left(\prod_{i\in A^j} \mathbb{1}_{\{x_i\geq 1\}}\right) \max\left\{ r_j + V^{t+1}\left(\boldsymbol{x} - \sum_{i\in A^j} \boldsymbol{e}_i\right), V^{t+1}(\boldsymbol{x})\right\} \\ &\qquad\qquad + \left(1 - \sum_{j\in\mathcal{J}} \lambda_j^t + \sum_{j\in\mathcal{J}} \lambda_j^t \left(1 - \prod_{i\in A^j} \mathbb{1}_{\{x_i\geq 1\}}\right)\right) V^{t+1}(\boldsymbol{x}) \\ &=\; V^{t+1}(\boldsymbol{x}) + \sum_{j\in\mathcal{J}} \lambda_j^t \left(\prod_{i\in A^j} \mathbb{1}_{\{x_i\geq 1\}}\right) \left[r_j - V^{t+1}(\boldsymbol{x}) + V^{t+1}\left(\boldsymbol{x} - \sum_{i\in A^j} \boldsymbol{e}_i\right)\right]^+, \end{aligned} \tag{1}$$

with the boundary condition that $V^{T+1} = 0$. In the dynamic program above, if we have a request for product j at time period t and we have capacities on all resources used by product j, then we have a choice to accept or reject the request. If we accept, then we generate a revenue of r_j and the state of the resources at the next time period is $\boldsymbol{x} - \sum_{i\in A^j} \boldsymbol{e}_i$. If we reject, then we do not generate revenue and the state of the resources at the next time period remains at $\boldsymbol{x}$. Also, if there is no request at time period t or there is a request for some product, but we do not have capacity on some resource used by the product, then we do not accept a request, in which case, the state of the resources at the next time period remains at $\boldsymbol{x}$. The second equality above follows by arranging the terms. Letting $\boldsymbol{C} = (C_i : i \in \mathcal{L})$ be the vector of initial resource capacities, the optimal total expected revenue is $V^1(\boldsymbol{C})$. By (1), given that the state of the resources at time period t is $\boldsymbol{x}$, if $r_j \geq V^{t+1}(\boldsymbol{x}) - V^{t+1}(\boldsymbol{x} - \sum_{i\in A^j} \boldsymbol{e}_i)$, then it is optimal to accept a request for product j as long as we have capacity on all resources used by product j. Here, we can view $V^{t+1}(\boldsymbol{x}) - V^{t+1}(\boldsymbol{x} - \sum_{i\in A^j} \boldsymbol{e}_i)$ as the opportunity cost of the capacities used by product j.

The size of the state space is $|\mathcal{Q}| = O\left(\prod_{i\in\mathcal{L}} C_i\right)$, which increases exponentially with the number of resources, making the computation of the optimal value functions intractable.

3. Approximate Policy

We construct approximations to the optimal value functions using a linear combination of basis functions. We use the following outline. In Section 3.1, we describe our basis functions. In Section 3.2, we show how to compute the coefficient of each basis function in the value function approximations. In Section 3.3, we show the performance guarantee for our approximate policy. In Section 3.4, we show that this performance guarantee is tight.

3.1 Requirements for Basis Functions

We approximate the optimal value function V^t using the value function approximation H^t. To construct the value function approximation H^t, we use a linear combination of basis functions. In particular, for each product j, we have a basis function $\varphi_j : \mathcal{Q} \to [0,1]$. In this case, the value function approximation H^t is given by

$$H^t(\boldsymbol{x}) \;=\; \sum_{j\in\mathcal{J}} \gamma_j^t\, \varphi_j(\boldsymbol{x}), \tag{2}$$

where $\mathscr{B} = \{\varphi_j : j \in \mathcal{J}\}$ is a prespecified collection of basis functions and $\{\gamma_j^t : j \in \mathcal{J},\ t \in \mathcal{T}\}$ are adjustable coefficients. To construct our approximate policy, we replace V^{t+1} on the right side of (1) with H^{t+1}. Therefore, given that the state of the resources at time period t is $\boldsymbol{x}$, our approximate policy accepts a request for product j as long as $r_j \geq H^{t+1}(\boldsymbol{x}) - H^{t+1}(\boldsymbol{x} - \sum_{i\in A^j} \boldsymbol{e}_i)$ and we have capacity on all resources used by product j. We refer to our basis functions as *availability-tracking basis functions* because we will impose the condition that $\varphi_j(\boldsymbol{x})$ takes the value of zero if the vector of resource capacities $\boldsymbol{x}$ does not provide enough availability to serve a request for product j. Below is the full definition of availability-tracking basis functions.

Definition 3.1 (Availability-Tracking Bases) The collection $\mathscr{B} = \{\varphi_j : j \in \mathcal{J}\}$ is called a collection of availability-tracking basis functions if it satisfies the following conditions.

(a) **Availability Tracking:** For each $j \in \mathcal{J}$ and $\boldsymbol{x} \in \mathcal{Q}$, we have $\varphi_j(\boldsymbol{x}) = 0$ whenever $x_i = 0$ for some $i \in A^j$. That is, $\varphi_j(\boldsymbol{x}) \leq \prod_{i\in A^j} \mathbb{1}_{\{x_i \geq 1\}}$.
(b) **Limited Dependence:** For each $j \in \mathcal{J}$ and $\boldsymbol{x} \in \mathcal{Q}$, the basis function $\varphi_j(\boldsymbol{x})$ only depends on $(x_i : i \in A^j)$. That is, for any $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{Q}$ with $x_i = y_i$ for all $i \in A^j$, we have $\varphi_j(\boldsymbol{x}) = \varphi_j(\boldsymbol{y})$.
(c) **Normalization:** For each $j \in \mathcal{J}$, we have $\varphi_j(\boldsymbol{C}) = 1$.

As discussed right before the definition, the range of φ_j is the interval $[0,1]$. The first property ensures that $\varphi_j(\boldsymbol{x})$ takes the value zero if the resource capacities $\boldsymbol{x}$ are not sufficient to serve a request for product j. The second property ensures that $\varphi_j(\boldsymbol{x})$ is independent of the capacities of the resources that are not used by product j. The third property ensures that $\varphi_j(\boldsymbol{x})$ is one when the resource capacities $\boldsymbol{x}$ are at their largest possible values. For example, the minimum basis function $\varphi_j(\boldsymbol{x}) = \min_{i\in A^j} \frac{x_i}{C_i}$ and the polynomial basis function $\varphi_j(\boldsymbol{x}) = \prod_{i\in A^j} \frac{x_i}{C_i}$ satisfy the three properties in the definition above. One can check that if $\varphi_j(\boldsymbol{x})$ is linear in $\boldsymbol{x}$, then it cannot satisfy the three properties. Therefore, linear basis functions are not availability-tracking.

Given a collection $\mathscr{B} = \{\varphi_j : j \in \mathcal{J}\}$ of availability-tracking basis functions, we define the *maximum scaled incremental contribution* $\Delta_{\mathscr{B}}$ from a unit of resource as

$$\Delta_{\mathscr{B}} = \max_{j\in\mathcal{J},\, i\in\mathcal{L}}\ \max_{\boldsymbol{x}\in\mathcal{Q}\,:\,x_i\geq 1} C_i \times \left(\, \varphi_j(\boldsymbol{x}) - \varphi_j(\boldsymbol{x} - \boldsymbol{e}_i)\,\right).$$

A collection with a smaller value of $\Delta_{\mathscr{B}}$ will yield an approximate policy with a better performance guarantee, but $\Delta_{\mathscr{B}}$ can never be smaller than one, as shown in the next lemma.

Lemma 3.2 (Bound on Change in the Value of Bases) *If* $\mathscr{B} = \{\varphi_j : j \in \mathcal{J}\}$ *is a collection of availability-tracking basis functions, then we must have* $\Delta_{\mathscr{B}} \geq 1$.

Proof: For any product j and resource $i \in A^j$, by parts (a) and (c) of Definition 3.1, we have $\varphi_j(\boldsymbol{C} - C_i \boldsymbol{e}_i) = 0$ and $\varphi_j(\boldsymbol{C}) = 1$. Therefore, a telescoping sum yields

$$1 = \varphi_j(\boldsymbol{C}) - \varphi_j(\boldsymbol{C} - C_i \boldsymbol{e}_i) = \sum_{h=1}^{C_i} \left[\varphi_j\left(\sum_{s \neq i} C_s \boldsymbol{e}_s + h\, \boldsymbol{e}_i \right) - \varphi_j\left(\sum_{s \neq i} C_s \boldsymbol{e}_s + (h-1)\, \boldsymbol{e}_i \right) \right] \leq \sum_{h=1}^{C_i} \frac{\Delta_{\mathscr{B}}}{C_i} = \Delta_{\mathscr{B}},$$

where the inequality holds because $\Delta_{\mathscr{B}} \geq C_i \times (\varphi_j(\sum_{s \neq i} C_s \boldsymbol{e}_s + h\, \boldsymbol{e}_i) - \varphi_j(\sum_{s \neq i} C_s \boldsymbol{e}_s + (h-1)\, \boldsymbol{e}_i))$ by the definition of $\Delta_{\mathscr{B}}$. $\square$

As shown in the next two examples, the lower bound in Lemma 3.2 is tight.

Example 3.3 (Minimum) Let $\varphi_j(\boldsymbol{x}) = \min_{i \in A^j} \frac{x_i}{C_i}$. It is simple to verify that $\mathscr{B} = \{\varphi_j : j \in \mathcal{J}\}$ satisfies the three properties in Definition 3.1, so $\Delta_{\mathscr{B}} \geq 1$ by Lemma 3.2. Moreover,

$$\varphi_j(\boldsymbol{x}) - \varphi_j(\boldsymbol{x} - \boldsymbol{e}_i) = \min_{\ell \in A^j} \left\{ \frac{x_\ell}{C_\ell} \right\} - \min_{\ell \in A^j} \left\{ \frac{x_\ell - \mathbb{1}_{\{\ell = i\}}}{C_\ell} \right\} = \begin{cases} \frac{1}{C_i} & \text{if } \frac{x_i}{C_i} = \min_{\ell \in A^j} \frac{x_\ell}{C_\ell} \\ \min_{\ell \in A^j} \frac{x_\ell}{C_\ell} - \frac{x_i - 1}{C_i} & \text{if } \frac{x_i}{C_i} > \min\limits_{\ell \in A^j} \frac{x_\ell}{C_\ell} \geq \frac{x_i - 1}{C_i} \\ 0 & \text{if } \frac{x_i - 1}{C_i} > \min_{\ell \in A^j} \frac{x_\ell}{C_\ell}. \end{cases}$$

The quantity in all three cases on the right side above is no larger than $1/C_i$, so that $C_i\,(\varphi_j(\boldsymbol{x}) - \varphi_j(\boldsymbol{x} - \boldsymbol{e}_i)) \leq 1$, in which case, we get $\Delta_{\mathscr{B}} \leq 1$. Thus, we must have $\Delta_{\mathscr{B}} = 1$.

Example 3.4 (Polynomial) Let $\varphi_j(\boldsymbol{x}) = \prod_{i \in A^j} \frac{x_i}{C_i}$. It is simple to verify that $\mathscr{B} = \{\varphi_j : j \in \mathcal{J}\}$ satisfies the three properties in Definition 3.1, so $\Delta_{\mathscr{B}} \geq 1$ by Lemma 3.2. Also,

$$\varphi_j(\boldsymbol{x}) - \varphi_j(\boldsymbol{x} - \boldsymbol{e}_i) = \left(\prod_{\ell \in A^j} \frac{x_\ell}{C_\ell} \right) - \left(\frac{x_i - 1}{C_i} \cdot \prod_{\ell \in A^j \setminus \{i\}} \frac{x_\ell}{C_\ell} \right) = \frac{1}{C_i} \prod_{\ell \in A^j \setminus \{i\}} \frac{x_\ell}{C_\ell} \leq \frac{1}{C_i},$$

where the last inequality follows because $x_\ell \leq C_\ell$. Therefore, we have $C_i\,(\varphi_j(\boldsymbol{x}) - \varphi_j(\boldsymbol{x} - \boldsymbol{e}_i)) \leq 1$, indicating that $\Delta_{\mathscr{B}} \leq 1$. So, we get $\Delta_{\mathscr{B}} = 1$.

By the discussion so far, if we set $\varphi_j(\boldsymbol{x}) = \min_{i \in A^j} \frac{x_i}{C_i}$ or $\varphi_j(\boldsymbol{x}) = \prod_{i \in A^j} \frac{x_i}{C_i}$ for all $j \in \mathcal{J}$, then the collection of basis functions $\mathscr{B} = \{\varphi_j : j \in \mathcal{J}\}$ is availability-tracking with $\Delta_{\mathscr{B}} = 1$. There are many other availability-tracking basis functions. In Appendix A, we give three results that allow us to construct rich collections of availability-tracking basis functions and guide us to choose a collection of availability-tracking basis functions. First, we show that if $\mathscr{B} = \{\varphi_j : j \in \mathcal{J}\}$ is a collection of availability-tracking basis functions and $f : [0,1] \to [0,1]$ is a non-decreasing and differentiable function with $f(0) = 0$ and $f(1) = 1$, then letting $\rho_j(\boldsymbol{x}) = f(\varphi_j(\boldsymbol{x}))$ for all $j \in \mathcal{J}$, the collection of basis functions $\mathscr{D} = \{\rho_j : j \in \mathcal{J}\}$ is also availability-tracking and $\Delta_{\mathscr{D}} \leq \Delta_{\mathscr{B}} \max_{a \in [0,1]} f'(a)$. For example, using this result with $\varphi_j(\boldsymbol{x}) = \prod_{i \in A^j} \frac{x_i}{C_i}$ and $f(a) = (1-e^{-a})/(1-e^{-1})$, it follows that the collection of basis functions $\mathscr{D} = \{\rho_j : j \in \mathcal{J}\}$ with

$$\rho_j(\boldsymbol{x}) = f(\varphi_j(\boldsymbol{x})) = \frac{1 - e^{-\prod_{i \in A^j} x_i/C_i}}{1 - e^{-1}}$$

is availability-tracking. It might be difficult to see at a first glance that the basis function above is availability-tracking. Second, we show that if, for each $i \in \mathcal{L}$, $g_i : [0,1] \to [0,1]$ is a non-decreasing and differentiable function with $g_i(0) = 0$ and $g_i(1) = 1$, then letting $\rho_j(\boldsymbol{x}) = \min_{i \in A^j} g_i(x_i/C_i)$ or $\rho_j(\boldsymbol{x}) = \prod_{i \in A^j} g_i(x_i/C_i)$ for all $j \in \mathcal{J}$, the collection of basis functions $\mathscr{D} = \{\rho_j : j \in \mathcal{J}\}$ is also availability-tracking and $\Delta_{\mathscr{D}} \leq \max_{i \in \mathcal{L}} \max_{a \in [0,1]} g_i'(a)$. For example, using this result with $g_i(a) = (1-e^{-a})/(1-e^{-1})$, it follows that the collection of basis functions $\{\rho_j : j \in \mathcal{J}\}$ with $\rho_j(\boldsymbol{x}) = \min_{i \in A^j}(1 - e^{-x_i/C_i})/(1-e^{-1})$ or $\rho_j(\boldsymbol{x}) = \prod_{i \in A^j}(1-e^{-x_i/C_i})/(1-e^{-1})$ is availability tracking. By using these two results, we can construct rich collections of availability-tracking basis functions. Third, Definition 3.1 does not require the availability-tracking basis functions to be non-decreasing, but we show that if $\mathscr{B} = \{\varphi_j : j \in \mathcal{J}\}$ is any collection of availability-tracking basis functions, then we can always construct a collection of non-decreasing availability-tracking basis functions $\mathscr{D} = \{\rho_j : j \in \mathcal{J}\}$ such that $\Delta_{\mathscr{D}} \leq \Delta_{\mathscr{B}}$. Since a collection of basis functions $\mathscr{B} = \{\varphi_j : j \in \mathcal{J}\}$ with a smaller value of $\Delta_{\mathscr{B}}$ will yield an approximate policy with a better performance guarantee, this result indicates that it is preferable to use non-decreasing basis functions. In Appendix A, we also give an example of a non-monotone basis function.

The discussion so far focuses on the basis functions $\{\varphi_j : j \in \mathcal{J}\}$ in the value function approximations in (2). Next, we discuss computing the coefficients $\{\gamma_j^t : j \in \mathcal{J},\ t \in \mathcal{T}\}$.

3.2 Tuning the Coefficients

To compute the coefficients $\{\gamma_j^t : j \in \mathcal{J},\ t \in \mathcal{T}\}$ in the value function approximations in (2), we use the following backward recursion over the time periods.

- **Initialization:** Let $\mathscr{B} = \{\varphi_j : j \in \mathcal{J}\}$ be any collection of availability-tracking basis functions and $\theta \geq \Delta_{\mathscr{B}}$ be a tuning parameter. Initialize $\gamma_j^{T+1} = 0$ for all $j \in \mathcal{J}$.
- **Coefficient Computation**: For each $t = T, T-1, \ldots, 1$, use the coefficients $\{\gamma_j^{t+1} : j \in \mathcal{J}\}$ to compute $\{\gamma_j^t : j \in \mathcal{J}\}$ as

$$\gamma_j^t \ = \ \lambda_j^t \left[r_j - \theta \sum_{i \in A^j} \frac{1}{C_i} \sum_{k \in \mathcal{J}} \mathbb{1}_{\{i \in A^k\}} \, \gamma_k^{t+1} \right]^+ + \ \gamma_j^{t+1}. \tag{3}$$

We shortly provide an intuitive interpretation for the algorithm above. The algorithm above allows us to compute $\{\gamma_j^t : j \in \mathcal{J},\ t \in \mathcal{T}\}$, which specifies the value function approximations $\{H^t : t \in \mathcal{T}\}$

given in (2). In our approximate policy, we make the decisions for the product request at time period t by replacing the optimal value function V^{t+1} on the right side of (1) with H^{t+1}. Thus, given that the state of the system at time period t is $\boldsymbol{x}$, if $r_j \geq H^{t+1}(\boldsymbol{x}) - H^{t+1}(\boldsymbol{x} - \sum_{i\in A^j} \boldsymbol{e}_i)$, then our approximate policy accepts a request for product j as long as we have capacity on all resources used by product j. Next, we give an intuitive interpretation for the coefficients $\{\gamma_j^t : j \in \mathcal{J},\ t \in \mathcal{T}\}$ and the backward recursion in (3) that we use to compute these coefficients.

To give an intuitive interpretation for the coefficients $\{\gamma_j^t : j \in \mathcal{J},\ t \in \mathcal{T}\}$, we note that the optimal total expected revenue over the selling horizon is given by $V^1(\boldsymbol{C})$. Therefore, $H^1(\boldsymbol{C})$ is our approximation to the optimal total expected revenue over the selling horizon. Since $\varphi_j(\boldsymbol{C}) = 1$ for all $j \in \mathcal{J}$ by part (c) of Definition 3.1, we have $H^1(\boldsymbol{C}) = \sum_{j\in\mathcal{J}} \gamma_j^1\, \varphi_j(\boldsymbol{C}) = \sum_{j\in\mathcal{J}} \gamma_j^1$, which implies that $\sum_{j\in\mathcal{J}} \gamma_j^1$ is our approximation to the optimal total expected revenue over the selling horizon. Thus, we can interpret γ_j^1 as our approximation to the optimal total expected revenue that we can extract from the requests for product j over the selling horizon. In this case, extending the argument intuitively to a generic time period, we will also interpret γ_j^t as our approximation to the optimal total expected revenue that we can extract from the requests for product j over time periods $t, t+1, \ldots, T$. Since γ_j^t is our approximation to the optimal total expected revenue that we can extract from the requests for product j over time periods $t, t+1, \ldots, T$, whereas γ_j^{t+1} is our approximation to the optimal total expected revenue that we can extract from the requests for product j over time periods $t+1, t+2, \ldots, T$, we naturally expect that $\gamma_j^t \geq \gamma_j^{t+1}$. Noting that $[a]^+ \geq 0$ for any $a \in \mathbb{R}$, by (3), we indeed have $\gamma_j^t \geq \gamma_j^{t+1}$.

To understand the intuitive idea behind the backward recursion in (3), recall that we view $V^{t+1}(\boldsymbol{x}) - V^{t+1}(\boldsymbol{x} - \sum_{i\in A^j} \boldsymbol{e}_i)$ as the opportunity cost of the capacities that are used by a request for product j that we accept at time period t. Using our value function approximations, we can approximate this opportunity cost by $H^{t+1}(\boldsymbol{x}) - H^{t+1}(\boldsymbol{x} - \sum_{i\in A^j} \boldsymbol{e}_i)$. Considering the value function approximation $H^{t+1}(\boldsymbol{x}) = \sum_{j\in\mathcal{J}} \gamma_j^{t+1}\, \varphi_j(\boldsymbol{x})$ with any collection of availability-tracking basis functions $\mathscr{B} = \{\varphi_j : j \in \mathcal{J}\}$, it is possible to show that we can upper bound the opportunity cost $H^{t+1}(\boldsymbol{x}) - H^{t+1}(\boldsymbol{x} - \sum_{i\in A^j} \boldsymbol{e}_i)$ as $H^{t+1}(\boldsymbol{x}) - H^{t+1}(\boldsymbol{x} - \sum_{i\in A^j} \boldsymbol{e}_i) \leq \theta \sum_{i\in A^j} \frac{1}{C_i} \sum_{k\in\mathcal{J}} \mathbb{1}_{\{i\in A^k\}}\, \gamma_k^{t+1}$, as long as $\theta \geq \Delta_{\mathscr{B}}$. We show this result in Lemma 3.7 when we analyze the performance of our approximate policy. Note that the upper bound $\theta \sum_{i\in A^j} \frac{1}{C_i} \sum_{k\in\mathcal{J}} \mathbb{1}_{\{i\in A^k\}}\, \gamma_k^{t+1}$ does not depend on the state of the system. Thus, we use $\theta \sum_{i\in A^j} \frac{1}{C_i} \sum_{k\in\mathcal{J}} \mathbb{1}_{\{i\in A^k\}}\, \gamma_k^{t+1}$ as a *state-independent* approximation to the opportunity cost of the capacities that are used by a request for product j that we accept at time period t. In this case, going back to (3), if we accept a request for product j at time period t, then we obtain a revenue of r_j and we approximate the opportunity cost of the capacities that we consume by using $\theta \sum_{i\in A^j} \frac{1}{C_i} \sum_{k\in\mathcal{J}} \mathbb{1}_{\{i\in A^k\}}\, \gamma_k^{t+1}$. We accept a request for product j only if the revenue from the product exceeds the opportunity cost of the capacities consumed by product j. Thus, $[r_j - \theta \sum_{i \in A^j} \frac{1}{C_i} \sum_{k \in \mathcal{J}} \mathbb{1}_{\{i \in A^k\}} \gamma_k^{t+1}]^+$ is our approximation to the net revenue contribution from a request for product j at time period t. By the discussion in the previous paragraph, γ_j^t is an approximation to the optimal total expected revenue from product j over time periods $t, t+1, \ldots, T$. Since we have a request for product j at time period t with probability λ_j^t, the recursion in (3) states the following. The approximation to the optimal total expected revenue from product j over time periods $t, t+1, \ldots, T$ is given by the approximation to the expected net revenue contribution from product j at time period t plus the approximation to the optimal total expected revenue from product j over time periods $t+1, t+2, \ldots, T$.

In the next section, we show that our approximate policy is guaranteed to obtain at least $1/(1+\theta L)$ fraction of the optimal total expected revenue. The proof of this performance guarantee involves two steps. In the first step, we use a linear program whose objective value provides an upper bound on the optimal total expected revenue. In particular, assuming that the total number of requests for each product takes on its expected value, this linear program finds the number of requests to accept for each product. The total expected number of requests for product j over the selling horizon is $\sum_{t \in \mathcal{T}} \lambda_j^t$. Letting $\Lambda_j = \sum_{t \in \mathcal{T}} \lambda_j^t$ for notational brevity and using the decision variable z_j to capture the number of requests for product j that we plan to accept over the selling horizon, the linear program has the form

$$\mathsf{Z}_{LP}^* \;=\; \max \left\{ \sum_{j \in \mathcal{J}} r_j \, z_j \;:\; \sum_{j \in \mathcal{J}} \mathbb{1}_{\{i \in A^j\}} \, z_j \le C_i \quad \forall \, i \in \mathcal{L}, \quad 0 \le z_j \le \Lambda_j \quad \forall \, j \in \mathcal{J} \right\}. \tag{4}$$

The objective function above accounts for the total revenue over the selling horizon. The first constraint ensures that the total capacity consumptions of the resources do not exceed the initial capacities. The second constraint ensures that the numbers of requests that we serve for the products do not exceed the expected demands. We show that we can use the coefficients $\{\gamma_j^t : j \in \mathcal{J}, \; t \in \mathcal{T}\}$ that are computed through (3) to construct a *feasible* solution to the dual of the linear program above and this feasible solution provides an objective value of at most $(1+\theta L) \sum_{j \in \mathcal{J}} \gamma_j^1$. Since the dual is a minimization problem, we get $(1+\theta L) \sum_{j \in \mathcal{J}} \gamma_j^1 \ge \mathsf{Z}_{LP}^* \ge V^1(\boldsymbol{C})$, where the second inequality uses the fact that the optimal objective value of the linear program is an upper bound on the optimal total expected revenue.

In the second step, we use backwards induction over the time periods to show that the total expected revenue obtained by our approximate policy is at least $\sum_{j \in \mathcal{J}} \gamma_j^1$. The intuition behind this result is that the recursion in (3) uses the upper bound $\theta \sum_{i \in A^j} \frac{1}{C_i} \sum_{k \in \mathcal{J}} \mathbb{1}_{\{i \in A^k\}} \gamma_k^{t+1}$ on the opportunity cost of the capacities used by each product j. So, intuitively speaking, γ_j^1 computed through the recursion in (3) turns out to be a pessimistic approximation to the optimal total expected revenue obtained from product j over the selling horizon. In this case, we will show that the total expected revenue obtained by our approximate policy is at least as large as the pessimistic approximation of the optimal total expected revenue $\sum_{j\in\mathcal{J}} \gamma_j^1$. Thus, letting APP be the total expected revenue obtained by our approximate policy, we have $\mathsf{APP} \geq \sum_{j\in\mathcal{J}} \gamma_j^1$. Using the chain of inequalities at the end of the previous paragraph, we get $\mathsf{APP} \geq \sum_{j\in\mathcal{J}} \gamma_j^1 \geq \frac{1}{1+\theta L} \mathsf{Z}_{LP}^* \geq \frac{1}{1+\theta L} V^1(\boldsymbol{C})$, establishing that the total expected revenue obtained by our approximate policy is guaranteed to be at least $1/(1+\theta L)$ fraction of the optimal total expected revenue.

In the next section, we follow the two steps discussed in the previous two paragraphs to show this performance guarantee for our approximate policy.

3.3 Performance Guarantee for the Approximate Policy

Recall that once we compute the coefficients $\{\gamma_j^t : j \in \mathcal{J},\ t \in \mathcal{T}\}$ by using (3), in our approximate policy, given that the state of the resources at time period t is $\boldsymbol{x}$, we accept a request for product j as long as $r_j \geq H^{t+1}(\boldsymbol{x}) - H^{t+1}(\boldsymbol{x} - \sum_{i\in A^j} \boldsymbol{e}_i)$ and we have capacity on all resources used by product j. To formally state our approximate policy, we use $u_j^{\mathsf{App},t} : \mathcal{Q} \to \{0,1\}$ to denote the decision function of our approximate policy at time period t. Given that the state of the resources at time period t is $\boldsymbol{x}$, we have $u_j^{\mathsf{App},t}(\boldsymbol{x}) = 1$ if our approximate policy accepts a request for product j at time period t. Otherwise, we have $u_j^{\mathsf{App},t}(\boldsymbol{x}) = 0$. Therefore, $u_j^{\mathsf{App},t}(\boldsymbol{x})$ is given by

$$u_j^{\mathsf{App},t}(\boldsymbol{x}) = \begin{cases} \prod_{i\in A^j} \mathbb{1}_{\{x_i \geq 1\}} & \text{if } \ r_j \geq H^{t+1}(\boldsymbol{x}) - H^{t+1}\left(\boldsymbol{x} - \sum_{i\in A^j} \boldsymbol{e}_i\right), \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

In the next theorem, we give a performance guarantee for this policy as a function of the tuning parameter θ in (3) and the maximum number of resources L used by a product.

Theorem 3.5 (Performance) *If the tuning parameter θ satisfies $\theta \geq \Delta_{\mathscr{B}}$, then the total expected revenue obtained by the approximate policy is at least $1/(1+\theta L)$ fraction of the optimal.*

We devote the rest of this section to giving a proof of Theorem 3.5, but before we go into the proof, we make three remarks on this theorem. First, to obtain the best performance guarantee, we need to choose the tuning parameter θ as small as possible and the smallest possible value of θ in the theorem is $\Delta_{\mathscr{B}}$. By Lemma 3.2, we have $\Delta_{\mathscr{B}} \geq 1$, but as shown in Examples 3.3 and 3.4, there are choices of basis functions under which $\Delta_{\mathscr{B}} = 1$. Therefore, working with these basis functions, we can choose the tuning parameter θ as one and obtain an approximate policy whose total expected revenue is at least $1/(1+L)$ fraction of the optimal total expected revenue. In many network revenue management problems arising in settings such as airlines and hotels, each product uses only a small number of resources. Therefore, even though the number of resources can be large, as long as the number of resources used by a product is uniformly bounded, we obtain a constant-factor approximation guarantee. Second, although we obtain the best performance guarantee by choosing θ at its smallest possible value of $\Delta_{\mathscr{B}}$, our computational experiments indicate that increasing θ beyond $\Delta_{\mathscr{B}}$ can improve the total expected revenue of the approximate policy. We view the tuning parameter θ as a knob that provides flexibility in the implementation of our approximate policy. Third, as is the case for most constant-factor approximation algorithms, the performance guarantee in Theorem 3.5 is a worst-case guarantee. In our computational experiments, we compare the total expected revenue obtained by our approximate policy with a computationally tractable upper bound on the optimal total expected revenue, and demonstrate that the approximate policy performs substantially better than its worst-case performance guarantee. Next, we turn to the proof of Theorem 3.5. The proof uses a sequence of lemmas.

In the next lemma, we show that $(1+\theta L)\sum_{j\in\mathcal{J}}\gamma_j^1$ is an upper bound on $V^1(\boldsymbol{C})$. The proof is based on using $\{\gamma_j^t : j \in \mathcal{J},\ t \in \mathcal{T}\}$ to construct a feasible solution to the dual of problem (4).

Lemma 3.6 (Upper Bound on the Optimal Total Expected Revenue) *If the coefficients* $\{\gamma_j^t : j \in \mathcal{J},\ t \in \mathcal{T}\}$ *are computed through* (3)*, then we have* $V^1(\boldsymbol{C}) \le (1+\theta L)\sum_{j\in\mathcal{J}}\gamma_j^1$.

Proof: It is a well-known result in the network revenue management literature that the optimal objective value of problem (4) is an upper bound on the optimal total expected revenue; see Bertsimas and Popescu (2003). Thus, we have $\mathsf{Z}_{LP}^* \ge V^1(\boldsymbol{C})$. Problem (4) is feasible and bounded because setting $z_j = 0$ for all $j \in \mathcal{J}$ provides a feasible solution and the decision variables have upper bounds. Thus, the optimal objective value of the dual of problem (4) is also Z_{LP}^*. Using the vectors of dual variables $\boldsymbol{\mu} = (\mu_i : i \in \mathcal{L})$ and $\boldsymbol{\sigma} = (\sigma_j : j \in \mathcal{J})$, the dual of problem (4) is $\min_{(\boldsymbol{\mu},\boldsymbol{\sigma})\in\mathbb{R}_+^{|\mathcal{L}|+|\mathcal{J}|}}\big\{\sum_{i\in\mathcal{L}} C_i\,\mu_i + \sum_{j\in\mathcal{J}}\Lambda_j\,\sigma_j \;:\; \sum_{i\in\mathcal{L}}\mathbb{1}_{\{i\in A^j\}}\mu_i + \sigma_j \ge r_j \quad \forall\, j\in\mathcal{J}\big\}$. The decision variable σ_j has a non-negative objective function coefficient, so it takes on its smallest possible value in an optimal solution. Writing the constraints in the dual problem as $\sigma_j \ge r_j - \sum_{i\in\mathcal{L}}\mathbb{1}_{\{i\in A^j\}}\mu_i$ for all $j\in\mathcal{J}$ and noting that the decision variable σ_j is non-negative, the smallest possible value of σ_j is $[r_j - \sum_{i\in\mathcal{L}}\mathbb{1}_{\{i\in A^j\}}\mu_i]^+$. In this case, replacing the decision variable σ_j by its value in an optimal solution, we can write the dual of problem (4) as

$$\mathsf{Z}_{LP}^* \;=\; \min_{\boldsymbol{\mu}\in\mathbb{R}_+^{|\mathcal{L}|}} \left\{ \sum_{i\in\mathcal{L}} C_i\,\mu_i + \sum_{j\in\mathcal{J}} \Lambda_j \left[r_j - \sum_{i\in\mathcal{L}} \mathbb{1}_{\{i\in A^j\}}\mu_i \right]^+ \right\}. \tag{6}$$

We define the solution $\hat{\boldsymbol{\mu}} = (\hat{\mu}_i : i\in\mathcal{L})$ to the problem above by setting $\hat{\mu}_i = \frac{\theta}{C_i}\sum_{k\in\mathcal{J}}\mathbb{1}_{\{i\in A^k\}}\gamma_k^1$ for all $i\in\mathcal{L}$. Noting (3), since $[a]^+ \ge 0$ for any $a\in\mathbb{R}$, we have $\gamma_j^1 \ge \gamma_j^2 \ge \ldots \ge \gamma_j^T \ge \gamma_j^{T+1} = 0$ for all

$j \in \mathcal{J}$, which implies that $\hat{\mu}_i \geq 0$ for all $i \in \mathcal{L}$. Thus, $\hat{\boldsymbol{\mu}}$ is a feasible solution to problem (6). In this case, the objective value provided by this solution is at least Z^*_{LP}, yielding

$$
\begin{aligned}
\mathsf{Z}^*_{LP} &\leq \sum_{i\in\mathcal{L}} C_i\,\hat{\mu}_i + \sum_{j\in\mathcal{J}} \Lambda_j \left[r_j - \sum_{i\in\mathcal{L}} \mathbb{1}_{\{i\in A^j\}} \hat{\mu}_i \right]^+ \\
&= \sum_{i\in\mathcal{L}} C_i \frac{\theta}{C_i} \sum_{k\in\mathcal{J}} \mathbb{1}_{\{i\in A^k\}}\, \gamma^1_k + \sum_{j\in\mathcal{J}} \Lambda_j \left[r_j - \sum_{i\in\mathcal{L}} \mathbb{1}_{\{i\in A^j\}} \frac{\theta}{C_i} \sum_{k\in\mathcal{J}} \mathbb{1}_{\{i\in A^k\}}\, \gamma^1_k \right]^+ \\
&= \theta \sum_{k\in\mathcal{J}} \sum_{i\in\mathcal{L}} \mathbb{1}_{\{i\in A^k\}}\, \gamma^1_k + \sum_{j\in\mathcal{J}} \sum_{t\in\mathcal{T}} \lambda^t_j \left[r_j - \theta \sum_{i\in A^j} \frac{1}{C_i} \sum_{k\in\mathcal{J}} \mathbb{1}_{\{i\in A^k\}}\, \gamma^1_k \right]^+ \\
&\leq \theta L \sum_{k\in\mathcal{J}} \gamma^1_k + \sum_{j\in\mathcal{J}} \sum_{t\in\mathcal{T}} \lambda^t_j \left[r_j - \theta \sum_{i\in A^j} \frac{1}{C_i} \sum_{k\in\mathcal{J}} \mathbb{1}_{\{i\in A^k\}}\, \gamma^{t+1}_k \right]^+ \\
&= \theta L \sum_{k\in\mathcal{J}} \gamma^1_k + \sum_{j\in\mathcal{J}} \sum_{t\in\mathcal{T}} [\gamma^t_j - \gamma^{t+1}_j] \;=\; (1+\theta L) \sum_{j\in\mathcal{J}} \gamma^1_j,
\end{aligned}
$$

where the second equality uses the fact that $\Lambda_j = \sum_{t\in\mathcal{T}} \lambda^t_j$, the second inequality uses the fact that $\sum_{i\in\mathcal{L}} \mathbb{1}_{\{i\in A^k\}} = |A^k| \leq L$ and $\gamma^1_k \geq \gamma^2_k \geq \ldots \geq \gamma^T_k \geq \gamma^{T+1}_k$ for all $k \in \mathcal{J}$, the third equality holds because $\gamma^t_j - \gamma^{t+1}_j = [r_j - \theta \sum_{i\in A^j} \frac{1}{C_i} \sum_{k\in\mathcal{J}} \mathbb{1}_{\{i\in A^k\}}\, \gamma^{t+1}_k]^+$ by (3) and the fourth equality follows by noting that $\sum_{t\in\mathcal{T}} [\gamma^t_j - \gamma^{t+1}_j] = \gamma^1_j$. Thus, we have $(1+\theta L) \sum_{j\in\mathcal{J}} \gamma^1_j \geq \mathsf{Z}^*_{LP}$, in which case, the desired result follows from the fact that $\mathsf{Z}^*_{LP} \geq V^1(\boldsymbol{C})$. □

In the next lemma, we bound the opportunity cost of the capacities that are consumed by product j under our value function approximations.

Lemma 3.7 (Bound on the Opportunity Cost) *For a collection of availability-tracking basis functions $\mathscr{B} = \{\varphi_j : j \in \mathcal{J}\}$, let $H^t(\boldsymbol{x}) = \sum_{k\in\mathcal{J}} \gamma^t_k\, \varphi_k(\boldsymbol{x})$, where the coefficients $\{\gamma^t_k : k \in \mathcal{J}\}$ satisfy $\gamma^t_k \geq 0$ for all $k \in \mathcal{J}$. Then, for each $j \in \mathcal{J}$ and $\boldsymbol{x} \in \mathcal{Q}$ such that $\boldsymbol{x} - \sum_{i\in A^j} \boldsymbol{e}_i \geq \boldsymbol{0}$, we have*

$$
H^t(\boldsymbol{x}) - H^t\left(\boldsymbol{x} - \sum_{i\in A^j} \boldsymbol{e}_i \right) \;\leq\; \Delta_{\mathscr{B}} \sum_{i\in A^j} \frac{1}{C_i} \sum_{k\in\mathcal{J}} \mathbb{1}_{\{i\in A^k\}} \gamma^t_k.
$$

Proof: We prove the result using induction on the cardinality of A^j. Consider the base case where $|A^j| = 1$ so that we have $A^j = \{i\}$ for some $i \in \mathcal{L}$. In this case, we get

$$
H^t(\boldsymbol{x}) - H^t(\boldsymbol{x} - \boldsymbol{e}_i) \;=\; \sum_{k\in\mathcal{J}} \mathbb{1}_{\{i\in A^k\}} \gamma^t_k \left(\varphi_k(\boldsymbol{x}) - \varphi_k(\boldsymbol{x} - \boldsymbol{e}_i) \right) \;\leq\; \frac{\Delta_{\mathscr{B}}}{C_i} \sum_{k\in\mathcal{J}} \mathbb{1}_{\{i\in A^k\}} \gamma^t_k, \tag{7}
$$

where the equality holds because $\varphi_k(\boldsymbol{x}) - \varphi_k(\boldsymbol{x} - \boldsymbol{e}_i) = 0$ whenever $i \notin A^k$ by part (b) of Definition 3.1 and the inequality follows from the definition of $\Delta_{\mathscr{B}}$. Thus, the base case holds.

Suppose that the result holds for any $|A^j| \le s$. Consider a case in which $|A^j| = s+1$, so $A^j = B \cup \{\ell\}$ for some $B \subseteq \mathcal{L}$ with $|B| = s$ and $\ell \in \mathcal{L}$ with $\ell \notin B$. Letting $\boldsymbol{y} = \boldsymbol{x} - \sum_{i \in B} \boldsymbol{e}_i$, we obtain

$$\begin{aligned}
H^t(\boldsymbol{x}) - H^t\left(\boldsymbol{x} - \sum_{i \in A^j} \boldsymbol{e}_i\right) &= H^t(\boldsymbol{x}) - H^t\left(\boldsymbol{x} - \sum_{i \in B} \boldsymbol{e}_i - \boldsymbol{e}_\ell\right) \\
&= H^t(\boldsymbol{x}) - H^t\left(\boldsymbol{x} - \sum_{i \in B} \boldsymbol{e}_i\right) + H^t\left(\boldsymbol{x} - \sum_{i \in B} \boldsymbol{e}_i\right) - H^t\left(\boldsymbol{x} - \sum_{i \in B} \boldsymbol{e}_i - \boldsymbol{e}_\ell\right) \\
&\le \Delta_{\mathscr{B}} \sum_{i \in B} \frac{1}{C_i} \sum_{k \in \mathcal{J}} \mathbb{1}_{\{i \in A^k\}} \gamma_k^t \;+\; H(\boldsymbol{y}) - H(\boldsymbol{y} - \boldsymbol{e}_\ell) \\
&\le \Delta_{\mathscr{B}} \sum_{i \in B} \frac{1}{C_i} \sum_{k \in \mathcal{J}} \mathbb{1}_{\{i \in A^k\}} \gamma_k^t \;+\; \frac{\Delta_{\mathscr{B}}}{C_\ell} \sum_{k \in \mathcal{J}} \mathbb{1}_{\{\ell \in A^k\}} \gamma_k^t \\
&= \Delta_{\mathscr{B}} \sum_{i \in A^j} \frac{1}{C_i} \sum_{k \in \mathcal{J}} \mathbb{1}_{\{i \in A^k\}} \gamma_k^t,
\end{aligned}$$

where the first inequality follows from the induction assumption, the second inequality follows from the base case, and the last equality is by the fact that $A^j = B \cup \{\ell\}$. $\square$

We obtain the term $\mathbb{1}_{\{i \in A^k\}}$ on the right side of (7) due to part (b) of Definition 3.1, which, in turn, yields the term $\mathbb{1}_{\{i \in A^k\}}$ on the right side of (3). Without this term, the upper bound in Lemma 3.6 would involve $(1 + \theta|\mathcal{L}|)$ rather than $(1 + \theta L)$. Next, we use this lemma to show that the total expected revenue obtained by our approximate policy is at least $\sum_{j \in \mathcal{J}} \gamma_j^1$. Let $U^t(\boldsymbol{x})$ be the total expected revenue obtained by our approximate policy over time periods $t, t+1, \ldots, T$ given that the state of the resources at time period t is $\boldsymbol{x}$. Noting the decision function for the approximate policy in (5), we can compute $\{U^t : t \in \mathcal{T}\}$ through the dynamic program

$$\begin{aligned}
U^t(\boldsymbol{x}) &= \sum_{j \in \mathcal{J}} \lambda_j^t \, u_j^{\mathsf{App},t}(\boldsymbol{x}) \left[r_j + U^{t+1}\left(\boldsymbol{x} - \sum_{i \in A^j} \boldsymbol{e}_i\right)\right] + \left(1 - \sum_{j \in \mathcal{J}} \lambda_j^t + \sum_{j \in \mathcal{J}} \lambda_j^t \,(1 - u_j^{\mathsf{App},t}(\boldsymbol{x}))\right) U^{t+1}(\boldsymbol{x}) \\
&= U^{t+1}(\boldsymbol{x}) + \sum_{j \in \mathcal{J}} \lambda_j^t \, u_j^{\mathsf{App},t}(\boldsymbol{x}) \left[r_j - U^{t+1}(\boldsymbol{x}) + U^{t+1}\left(\boldsymbol{x} - \sum_{i \in A^j} \boldsymbol{e}_i\right)\right], \qquad (8)
\end{aligned}$$

with the boundary condition that $U^{T+1} = 0$. In the dynamic program above, we have a request for product j at time period t with probability λ_j^t, in which case, if we have $u_j^{\mathsf{App},t}(\boldsymbol{x}) = 1$ so that the approximate policy accepts this request, then we obtain a revenue of r_j and the state of the resources at the next time period is $\boldsymbol{x} - \sum_{i \in A^j} \boldsymbol{e}_i$. If there is a request for product j at time period t, but we have $u_j^{\mathsf{App},t}(\boldsymbol{x}) = 0$, then the state of the resources at the next time period remains at $\boldsymbol{x}$. With probability $1 - \sum_{j \in \mathcal{J}} \lambda_j^t$, there is no request, in which case, the state of the resources at the next time period also remains at $\boldsymbol{x}$. The second equality above is by arranging the terms.

The total expected revenue obtained by our approximate policy is $U^1(\boldsymbol{C})$. In the next lemma, we show that this total expected revenue is at least $\sum_{j \in \mathcal{J}} \gamma_j^1$.

Lemma 3.8 (Lower Bound on Performance) *If the tuning parameter θ satisfies $\theta \geq \Delta_{\mathscr{B}}$ and the coefficients $\{\gamma_j^t : j \in \mathcal{J},\ t \in \mathcal{T}\}$ are computed through* (3)*, then we have $U^1(\boldsymbol{C}) \geq \sum_{j \in \mathcal{J}} \gamma_j^1$.*

Proof: We use induction over the time periods to show that $U^t(\boldsymbol{x}) \geq H^t(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{Q}$ and $t \in \mathcal{T}$, where $H^t(\boldsymbol{x}) = \sum_{j \in \mathcal{J}} \gamma_j^t \varphi_j(\boldsymbol{x})$ with $\{\gamma_j^t : j \in \mathcal{J},\ t \in \mathcal{T}\}$ computed through (3). Consider the base case at time period $T+1$. Since $U^{T+1} = 0$ and $\gamma_j^{T+1} = 0$, the base case holds. Suppose that the result holds at time period $t+1$, so $U^{t+1}(\boldsymbol{x}) \geq H^{t+1}(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{Q}$. On the right side of (8), the coefficients of $U^{t+1}(\boldsymbol{x})$ and $U^{t+1}(\boldsymbol{x} - \sum_{i \in A^j} \boldsymbol{e}_i)$ are, respectively, $1 - \sum_{j \in \mathcal{J}} \lambda_j^t u_j^{\mathsf{App},t}(\boldsymbol{x})$ and $\lambda_j^t u_j^{\mathsf{App},t}(\boldsymbol{x})$, which are non-negative. Since $U^{t+1} \geq H^{t+1}$ by the induction assumption, replacing U^{t+1} on the right side of (8) with H^{t+1}, the right side of (8) gets smaller. Thus, we have

$$
\begin{aligned}
U^t(\boldsymbol{x}) &\geq H^{t+1}(\boldsymbol{x}) + \sum_{j \in \mathcal{J}} \lambda_j^t u_j^{\mathsf{App},t}(\boldsymbol{x}) \left[r_j - H^{t+1}(\boldsymbol{x}) + H^{t+1}\left(\boldsymbol{x} - \sum_{i \in A^j} \boldsymbol{e}_i \right) \right] \\
&= H^{t+1}(\boldsymbol{x}) + \sum_{j \in \mathcal{J}} \lambda_j^t \left(\prod_{i \in A^j} \mathbb{1}_{\{x_i \geq 1\}} \right) \left[r_j - H^{t+1}(\boldsymbol{x}) + H^{t+1}\left(\boldsymbol{x} - \sum_{i \in A^j} \boldsymbol{e}_i \right) \right]^+ \\
&\geq H^{t+1}(\boldsymbol{x}) + \sum_{j \in \mathcal{J}} \lambda_j^t \left(\prod_{i \in A^j} \mathbb{1}_{\{x_i \geq 1\}} \right) \left[r_j - \theta \sum_{i \in A^j} \frac{1}{C_i} \sum_{k \in \mathcal{J}} \mathbb{1}_{\{i \in A^k\}} \gamma_k^{t+1} \right]^+ \\
&= H^{t+1}(\boldsymbol{x}) + \sum_{j \in \mathcal{J}} \left(\prod_{i \in A^j} \mathbb{1}_{\{x_i \geq 1\}} \right) \left(\gamma_j^t - \gamma_j^{t+1} \right) \\
&\geq H^{t+1}(\boldsymbol{x}) + \sum_{j \in \mathcal{J}} \varphi_j(\boldsymbol{x}) \left(\gamma_j^t - \gamma_j^{t+1} \right) \;=\; H^t(\boldsymbol{x}).
\end{aligned} \tag{9}
$$

Here, the first equality follows by the definition of the decision function in (5) as we have $u_j^{\mathsf{App},t}(\boldsymbol{x}) = 1$ if and only if $\prod_{i \in A^j} \mathbb{1}_{\{x_i \geq 1\}} = 1$ and $r_j - H^{t+1}(\boldsymbol{x}) + H^{t+1}(\boldsymbol{x} - \sum_{i \in A^j} \boldsymbol{e}_i) \geq 0$. The second inequality follows from Lemma 3.7 and the fact that $\theta \geq \Delta_{\mathscr{B}}$. The second equality uses the definition of γ_j^t in (3). The third inequality follows from part (a) of Definition 3.1, along with the fact that $\gamma_j^t - \gamma_j^{t+1} \geq 0$. The last equality holds because we have $H^{t+1}(\boldsymbol{x}) = \sum_{j \in \mathcal{J}} \gamma_j^{t+1} \varphi_j(\boldsymbol{x})$. So, $U^t(\boldsymbol{x}) \geq H^t(\boldsymbol{x})$, completing the induction argument. This inequality at $t = 1$ and $\boldsymbol{x} = \boldsymbol{C}$, along with part (c) of Definition 3.1, yields $U^1(\boldsymbol{C}) \geq H^1(\boldsymbol{C}) = \sum_{j \in \mathcal{J}} \gamma_j^1 \varphi_j(\boldsymbol{C}) = \sum_{j \in \mathcal{J}} \gamma_j^1$. $\square$

Note that due to part (a) of Definition 3.1, we can use $\varphi_j(\boldsymbol{x})$ in the last inequality in (9) as a lower bound substitute on the feasibility condition $\prod_{i \in A^j} \mathbb{1}_{\{x_i \geq 1\}}$ for accepting a request for product j. The proof of Theorem 3.5 directly follows by combining Lemmas 3.6 and 3.8.

Proof of Theorem 3.5: The optimal total expected revenue is $V^1(\boldsymbol{C})$, whereas the total expected revenue obtained by our approximate policy is $U^1(\boldsymbol{C})$. In this case, by Lemmas 3.6 and 3.8, we have $U^1(\boldsymbol{C}) \geq \sum_{j \in \mathcal{J}} \gamma_j^1 \geq V^1(\boldsymbol{C})/(1 + \theta L)$. $\square$

3.4 Tightness of the Analysis

In this section, we give a class of problem instances to demonstrate that the performance guarantee in Theorem 3.5 is tight for any L. Fix two positive integers K and β. We consider a problem instance with K resources and $K+1$ products. We index the resources by $\mathcal{L}=\{1,\ldots,K\}$ and the products by $\mathcal{J}=\{1,\ldots,K,K+1\}$. The initial capacities of the resources are $C_i=\beta+1$ for all $i\in\mathcal{L}$. For $j=1,\ldots,K$, product j uses only one unit of resource j. Product $K+1$ uses all of the K resources. That is, $A^j=\{j\}$ for all $j=1,\ldots,K$ and $A^{K+1}=\mathcal{L}$. Note that $L=\max_{j\in\mathcal{J}}|A^j|=K$. The revenues of the products are given by $r_j=\frac{1}{\beta+1}\left(1-\frac{1}{\beta}\right)$ for all $j=1,\ldots,K$ and $r_{K+1}=1$. There are $K\beta+1$ time periods in the selling horizon indexed by $\mathcal{T}=\{1,\ldots,K\beta+1\}$. For $j=1,\ldots,K$, the requests for product j arrive only at time periods $(j-1)\beta+1,(j-1)\beta+2,\ldots,j\beta$. At the last time period, we have a request of product $K+1$. In particular, for $j=1,\ldots,K$, we have

$$\lambda_j^t=\begin{cases}1 & \text{if}\quad (j-1)\,\beta+1\le t\le j\beta,\\ 0 & \text{otherwise,}\end{cases}\qquad\text{and}\qquad \lambda_{K+1}^t=\begin{cases}1 & \text{if}\quad t=K\beta+1,\\ 0 & \text{otherwise.}\end{cases}$$

Even if we accept all of the product requests, we do not run out of the capacities of any of the resources. Thus, the optimal policy accepts all requests, in which case, the total expected revenue of the optimal policy is $\mathsf{OPT}=K\beta\,r_1+r_{K+1}=\frac{K\beta}{\beta+1}\left(1-\frac{1}{\beta}\right)+1$.

We consider our approximate policy using the collection of availability-tracking basis functions $\mathscr{B}=\{\varphi_j:j\in\mathcal{J}\}$ with $\varphi_j(\boldsymbol{x})=\min_{i\in A^j}\frac{x_i}{C_i}$ for all $j\in\mathcal{J}$. In Example 3.3, we show that we have $\Delta_{\mathscr{B}}=1$ for this collection of basis functions. Therefore, we can choose $\theta=1$ in the recursion in (3). We proceed to computing the coefficients $\{\gamma_j^t:j\in\mathcal{J},\ t\in\mathcal{T}\}$. Since $\gamma_{K+1}^{K\beta+2}=0$, by (3), we have $\gamma_{K+1}^{K\beta+1}=\lambda_{K+1}^{K\beta+1}r_{K+1}=1$. Also, noting that $\lambda_{K+1}^t=0$ for all $t\in\mathcal{T}\setminus\{K\beta+1\}$, we have $\gamma_{K+1}^1=\gamma_{K+1}^2=\ldots=\gamma_{K+1}^{K\beta+1}=1$. For $j=1,\ldots,K$, by (3), we have

$$\gamma_j^t\;=\;\lambda_j^t\left[r_j-\frac{\gamma_j^{t+1}+\gamma_{K+1}^{t+1}}{C_j}\right]^+\;+\;\gamma_j^{t+1}\;=\;\lambda_j^t\left[r_j-\frac{\gamma_j^{t+1}+\gamma_{K+1}^{t+1}}{\beta+1}\right]^+\;+\;\gamma_j^{t+1}.$$

Since $\gamma_j^{K\beta+2}=0$, noting that $\lambda_j^{K\beta+1}=0$, we obtain $\gamma_j^{K\beta+1}=0$ by the recursion above. Also, since $\gamma_{K+1}^1=\gamma_{K+1}^2=\ldots=\gamma_{K+1}^{K\beta+1}=1$, for all $t\in\mathcal{T}\setminus\{K\beta+1\}$, we have $r_j-\frac{\gamma_j^{t+1}+\gamma_{K+1}^{t+1}}{\beta+1}\le r_j-\frac{\gamma_{K+1}^{t+1}}{\beta+1}=\frac{1}{\beta+1}\left(1-\frac{1}{\beta}\right)-\frac{1}{\beta+1}\le 0$. Therefore, we get $\left[r_j-\frac{\gamma_j^{t+1}+\gamma_{K+1}^{t+1}}{\beta+1}\right]^+=0$ for all $t\in\mathcal{T}\setminus\{K\beta+1\}$, in which case, noting the recursion above, we have $\gamma_j^1=\gamma_j^2=\ldots=\gamma_j^{K\beta+1}=0$. Thus, by the discussion in this paragraph, we have $\gamma_{K+1}^t=1$ for all $t\in\mathcal{T}$. Also, for $j=1,\ldots,K$, we have $\gamma_j^t=0$ for all $t\in\mathcal{T}$. In this case, we get $H^t(\boldsymbol{x})=\sum_{j\in\mathcal{J}}\gamma_j^t\min_{i\in A^j}\frac{x_i}{C_i}=\gamma_{K+1}^t\min_{i\in\mathcal{L}}\frac{x_i}{C_i}=\frac{1}{\beta+1}\min_{i\in\mathcal{L}}x_i$ for all $t\in\mathcal{T}$.

Consider the decisions made by our approximate policy. Initially, the capacities of the resources are $C_i=\beta+1$ for all $i\in\mathcal{L}$. At the first time period, we have a request for product 1. Since $r_1=\frac{1}{\beta+1}\left(1-\frac{1}{\beta}\right)<\frac{1}{\beta+1}=H^2(\boldsymbol{C})-H^2(\boldsymbol{C}-\boldsymbol{e}_1)$, we do not accept the request. Since $r_1=r_2=\ldots=r_K$, a similar argument shows that we reject all of the requests at time periods $1, 2, \ldots, K\beta$. However, we accept the request for product $K+1$ at the last time period, because $H^{K\beta+2}(\boldsymbol{x}) = 0$, and thus, $r_{K+1} = 1 > 0 = H^{K\beta+2}(\boldsymbol{C}) - H^{K\beta+2}(\boldsymbol{C} - \sum_{i \in A^{K+1}} \boldsymbol{e}_i)$. Therefore, the total expected revenue from our approximate policy is $\mathsf{APP} = r_{K+1} = 1$. So, the ratio between the total expected revenues of our approximate policy and the optimal policy is $\frac{\mathsf{APP}}{\mathsf{OPT}} = 1/\left(1 + \frac{K\beta}{\beta+1}\left(1 - \frac{1}{\beta}\right)\right)$. If we choose β arbitrarily large, then $\frac{\mathsf{APP}}{\mathsf{OPT}}$ becomes arbitrarily close to $1/(1+K)$, which is equal to $1/(1+L)$.

4. Extensions

We extend our approximate policy to the cases in which the customers choose among the offered products, and a product can use more than one unit of the capacity of a resource. We also discuss leveraging a linear program to build value function approximations.

4.1 Customer Choice Behavior

In the model in Section 2, each customer enters the system with a request for a particular product. We decide whether to accept or reject the request for this product. In this section, we extend our model and performance guarantee to a case in which we offer a subset of products to each arriving customer, and the customer chooses among the offered products or decides to leave without a purchase. Therefore, the customer does not arrive with a request for a particular product, and the product that the customer ends up choosing may depend on the subset of products that we offer. The notation that we use closely follows the one introduced in Section 2. We only discuss the additional notation that we need. If we offer the subset $S \subseteq \mathcal{J}$ of products to a customer arriving at time period t, then the customer chooses product $j \in S$ with probability $\phi_j^t(S)$. Naturally, we have $\phi_j^t(S) = 0$ for all $j \notin S$. Note that the choices of the customers at different time periods may be governed by different purchase probabilities. Therefore, we allow nonstationarities in the choice process of the customers. We refer to a subset of products that we offer to the customers as an assortment. We use $\mathcal{F} \subseteq 2^{\mathcal{J}}$ to denote the set of feasible assortments that we can offer to an arriving customer. We impose the following mild assumption on the choice probabilities and the set of feasible assortments that we can offer to the customers.

Assumption 4.1 (Substitutability and Feasibility) *For all $t \in \mathcal{T}$, $S \in \mathcal{F}$, $j \in S$, and $k \notin S$, we have $\phi_j^t(S) \geq \phi_j^t(S \cup \{k\})$. Also, if $S \in \mathcal{F}$, then we have $R \in \mathcal{F}$ for all $R \subseteq S$.*

The first part of the assumption ensures that if we introduce an additional product into the assortment S, then the choice probability of a product that is already in the assortment S does not increase. This property holds for all choice models that are based on the random utility maximization principle. The second part of the assumption ensures that if we remove products from a feasible assortment, then the assortment remains feasible. To formulate the problem as a dynamic program, we let $V^t(\boldsymbol{x})$ be the optimal total expected revenue over time periods $t, t+1, \ldots, T$ given that the capacities of the resources at time period t is $\boldsymbol{x}$. We can compute the optimal value functions $\{V^t(\boldsymbol{x}) : \boldsymbol{x} \in \mathcal{Q},\ t \in \mathcal{T}\}$ using the dynamic program

$$\begin{aligned} V^t(\boldsymbol{x}) \;&=\; \max_{S \in \mathcal{F}} \Bigg\{ \sum_{j \in \mathcal{J}} \phi_j^t(S) \left(\prod_{i \in A^j} \mathbb{1}_{\{x_i \geq 1\}} \right) \left[r_j + V^{t+1} \left(\boldsymbol{x} - \sum_{i \in A^j} \boldsymbol{e}_i \right) \right] \\ &\qquad\qquad + \left(1 - \sum_{j \in \mathcal{J}} \phi_j^t(S) + \sum_{j \in \mathcal{J}} \phi_j^t(S) \left(1 - \prod_{i \in A^j} \mathbb{1}_{\{x_i \geq 1\}} \right) \right) V^{t+1}(\boldsymbol{x}) \Bigg\} \\ &=\; V^{t+1}(\boldsymbol{x}) + \max_{S \in \mathcal{F}} \left\{ \sum_{j \in \mathcal{J}} \phi_j^t(S) \left(\prod_{i \in A^j} \mathbb{1}_{\{x_i \geq 1\}} \right) \left(r_j - V^{t+1}(\boldsymbol{x}) + V^{t+1} \left(\boldsymbol{x} - \sum_{i \in A^j} \boldsymbol{e}_i \right) \right) \right\}, \quad (10) \end{aligned}$$

with the boundary condition that $V^{T+1} = 0$. If $|A^j| = 1$ for all $j \in \mathcal{J}$ so that each product uses exactly one resource, then our model becomes equivalent to the one in Golrezaei et al. (2014). In Golrezaei et al. (2014), there are multiple customer types, but multiple customer types do not bring any complication. Under multiple customer types, the result of the max operator above depends on the customer type and we simply take an expectation over the arriving customer type.

In the first equality in (10), if we offer the assortment S at time period t, then the arriving customer chooses product j with probability $\phi_j^t(S)$. If we have sufficient resource capacities to serve product j, so that $\prod_{i \in A^j} \mathbb{1}_{\{x_i \geq 1\}} = 1$, then the customer purchases product j, in which case, we generate a revenue of r_j and the state of the resources at the next time period is $\boldsymbol{x} - \sum_{i \in A^j} \boldsymbol{e}_i$. On the other hand, the arriving customer does not choose any product and decides to leave without a purchase with probability $1 - \sum_{j \in \mathcal{J}} \phi_j^t(S)$, in which case, the state of the resources at the next time period remains at $\boldsymbol{x}$. Lastly, the arriving customer chooses product j with probability $\phi_j^t(S)$, but if we do not have sufficient resource capacities to serve product j, so that $\prod_{i \in A^j} \mathbb{1}_{\{x_i \geq 1\}} = 0$, then the customer leaves without a purchase, in which case, the state of the resources at the next time period remains at $\boldsymbol{x}$ as well. In (10), we allow offering a product for which we lack sufficient resource capacities to serve. If the customer ends up choosing such a product, then the customer leaves without a purchase. Offering a product for which we lack sufficient resource capacities to serve may not sound realistic, but it is simple to argue that there exists an optimal policy that never offers such a product anyway. To establish this result, note that in the optimal solution to the second maximization problem in (10), if we attempt to offer products for which we do not have enough resource capacity to serve, then we can drop all such products from the assortment, along with each product j such that $r_j - V^{t+1}(\boldsymbol{x}) + V^{t+1}(\boldsymbol{x} - \sum_{i \in A^j} \boldsymbol{e}_i) < 0$, in which case, by Assumption 4.1, the choice probabilities of all remaining products in the assortment do not decrease, yielding another assortment that provides an objective value that is as large as the original one.

Similar to our earlier model, computing the optimal value functions $\{V^t : t \in \mathcal{T}\}$ is intractable. We use a value function approximation of the form $H^t(\boldsymbol{x}) = \sum_{j\in\mathcal{J}} \gamma_j^t \varphi_j(\boldsymbol{x})$, where $\mathscr{B} = \{\varphi_j : j \in \mathcal{J}\}$ is a collection of availability-tracking basis functions. We compute the coefficients $\{\gamma_j^t : j \in \mathcal{J},\ t \in \mathcal{T}\}$ in the value function approximations using a slight variation of our earlier algorithm.

- **Initialization:** Let $\mathscr{B} = \{\varphi_j : j \in \mathcal{J}\}$ be any collection of availability-tracking basis functions and $\theta \geq \Delta_{\mathscr{B}}$ be a tuning parameter. Initialize $\gamma_j^{T+1} = 0$ for all $j \in \mathcal{J}$.
- **Coefficient Computation**: For each $t = T, T-1, \ldots, 1$, use the coefficients $\{\gamma_j^{t+1} : j \in \mathcal{J}\}$ to compute the assortment $\hat{S}^t \in \mathcal{F}$ at time period t as

$$\hat{S}^t = \arg\max_{S\in\mathcal{F}} \left\{ \sum_{j\in\mathcal{J}} \phi_j^t(S) \left(r_j - \theta \sum_{i\in A^j} \frac{1}{C_i} \sum_{k\in\mathcal{J}} \mathbb{1}_{\{i\in A^k\}} \gamma_k^{t+1} \right) \right\}. \tag{11}$$

Then, use the coefficients $\{\gamma_j^{t+1} : j \in \mathcal{J}\}$ and the assortment $\hat{S}^t$ computed above to compute $\{\gamma_j^t : j \in \mathcal{J}\}$ as

$$\gamma_j^t \; = \; \phi_j^t(\hat{S}^t) \left(r_j - \theta \sum_{i\in A^j} \frac{1}{C_i} \sum_{k\in\mathcal{J}} \mathbb{1}_{\{i\in A^k\}} \gamma_k^{t+1} \right) + \; \gamma_j^{t+1}. \tag{12}$$

The algorithm above specifies the coefficients $\{\gamma_j^t : j \in \mathcal{J},\ t \in \mathcal{T}\}$, which, in turn, specify the approximate value functions $\{H^t : t \in \mathcal{T}\}$.

Given that the state of the resources at time period t is $\boldsymbol{x}$, we solve the maximization problem on the right side of the second equality in (10) to find the optimal assortment to offer. We construct our approximate policy by replacing V^{t+1} in this problem with H^{t+1}. Thus, given that the state of the resources at time period t is $\boldsymbol{x}$, our approximate policy offers the assortment

$$S^{\mathsf{App},t}(\boldsymbol{x}) \; = \; \arg\max_{S\in\mathcal{F}} \left\{ \sum_{j\in\mathcal{J}} \phi_j^t(S) \left(\prod_{i\in A^j} \mathbb{1}_{\{x_i\geq 1\}} \right) \left(r_j - H^{t+1}(\boldsymbol{x}) + H^{t+1}\left(\boldsymbol{x} - \sum_{i\in A^j} \boldsymbol{e}_i \right) \right) \right\}. \tag{13}$$

An optimal solution to the problem above can be viewed as the decision function of our approximate policy under customer choice behavior. By the next theorem, our approximate policy enjoys the same performance guarantee as in Theorem 3.5. The proof of this theorem uses a technique similar to the one in Section 3.3. We defer the proof to Appendix B.

Theorem 4.2 (Performance under Choice) *If the choice probabilities and the feasible assortments satisfy Assumption 4.1 and the tuning parameter* θ *satisfies* $\theta \geq \Delta_{\mathscr{B}}$*, then the total expected revenue obtained by the approximate policy is at least* $1/(1+\theta L)$ *fraction of the optimal.*

4.2 Multiple Units of Capacity Consumption

In this section, we extend our approach to allow products to use multiple units of a resource. Once again, the notation that we use closely follows the one introduced in Section 2. We only discuss the additional notation that we need. For each product j and resource i, we use a_{ij} to denote the number of units of resource i used by product j. In our earlier model, we have $a_{ij} \in \{0,1\}$ for all $i \in \mathcal{L}$, $j \in \mathcal{J}$. In this section, we consider the case where a_{ij} can be any non-negative integer. As before, $A^j = \{i \in \mathcal{L} : a_{ij} \geq 1\}$ denotes the set of resources used by product j, and $L = \max_{j \in \mathcal{J}} |A^j|$ denotes the maximum number of resources used by a product. We use $m_i = \max_{j \in \mathcal{J}} a_{ij}$ to denote the maximum number of units of resource i that is used by any product. Without loss of generality, we assume that the initial capacity of each resource i satisfies $C_i \geq m_i$. Otherwise, there is a product that uses more units than the initial capacity of a resource, in which case, we can drop such a product. To find the optimal policy, we can use a dynamic program that is similar to the one in (1). All we need to do is to replace all occurrences of $\prod_{i \in A^j} \mathbb{1}_{\{x_i \geq 1\}}$ with $\prod_{i \in A^j} \mathbb{1}_{\{x_i \geq a_{ij}\}}$ and all occurrences of $\sum_{i \in A^j} \boldsymbol{e}_i$ with $\sum_{i \in A^j} a_{ij}\, \boldsymbol{e}_i$ in the dynamic program.

We modify our basis functions as follows. For each product $j \in \mathcal{J}$, let $\mathtt{G}_j : \mathcal{Q} \to \mathcal{Q}$ be a mapping such that for each $\boldsymbol{x} \in \mathcal{Q}$, $\mathtt{G}_j(\boldsymbol{x}) = \big(x_i \mathbb{1}_{\{x_i \geq a_{ij}\}} : i \in \mathcal{L}\big)$. Thus, $\mathtt{G}_j(\boldsymbol{x})$ leaves the i-th component of $\boldsymbol{x}$ unchanged when the value of the component exceeds the amount of resource i consumed by product j; otherwise, $\mathtt{G}_j(\boldsymbol{x})$ sets the component to zero. For a collection of availability-tracking basis functions $\mathscr{B} = \{\varphi_j : j \in \mathcal{J}\}$, we use the value function approximation H^t given by

$$H^t(\boldsymbol{x}) = \sum_{j \in \mathcal{J}} \gamma_j^t \, \varphi_j \left(\mathtt{G}_j(\boldsymbol{x}) \right). \tag{14}$$

We compute the coefficients $\{\gamma_j^t : j \in \mathcal{J},\ t \in \mathcal{T}\}$ in the value function approximation above using the following algorithm.

- **Initialization:** Let $\mathscr{B} = \{\varphi_j : j \in \mathcal{J}\}$ be any collection of availability-tracking basis functions and $\theta \geq \Delta_{\mathscr{B}}$ be a tuning parameter. Initialize $\gamma_j^{T+1} = 0$ for all $j \in \mathcal{J}$.
- **Coefficient Computation**: For each $t = T, T-1, \ldots, 1$, use the coefficients $\{\gamma_j^{t+1} : j \in \mathcal{J}\}$ to compute $\{\gamma_j^t : j \in \mathcal{J}\}$ as

$$\gamma_j^t \;=\; \lambda_j^t \left[r_j - \theta \sum_{i \in A^j} \frac{2\, m_i - 1}{C_i} \sum_{k \in \mathcal{J}} \mathbb{1}_{\{i \in A^k\}}\, \gamma_k^{t+1} \right]^+ +\; \gamma_j^{t+1}. \tag{15}$$

To construct our approximate policy, we use the decision function in (5) after replacing $\prod_{i \in A^j} \mathbb{1}_{\{x_i \geq 1\}}$ with $\prod_{i \in A^j} \mathbb{1}_{\{x_i \geq a_{ij}\}}$ and $\sum_{i \in A^j} \boldsymbol{e}_i$ with $\sum_{i \in A^j} a_{ij}\, \boldsymbol{e}_i$. This decision function provides the decisions made by our approximate policy given that the state of the resources at time period t is $\boldsymbol{x}$. The following theorem gives a performance guarantee for our approximate policy when a product can consume multiple units of a resource. The proof is in Appendix C.

Theorem 4.3 (Performance under Multiple Units Consumption) *Let $M = \max_{i \in \mathcal{L}, j \in \mathcal{J}} a_{ij}$ be the maximum number of units of a resource used by a product. If the tuning parameter θ satisfies $\theta \geq \Delta_{\mathscr{B}}$, then the total expected revenue obtained by the approximate policy is at least $1/(1+\theta(2M-1)L)$ fraction of the optimal.*

To intuitively see the origin of the term $2M-1$, assume that all products use one unit of a resource. If the capacity of the resource goes down to one, then we can accept at most one unit of any product. In contrast, assume that all products use M units of a resource. If the capacity of the resource goes down to $2M-1$, then we can accept at most one unit of any product.

4.3 Leveraging a Linear Programming Approximation

Noting that $\Lambda_j = \sum_{t \in \mathcal{T}} \lambda_j^t$ in the second constraint in problem (4) corresponds to the total expected number of requests for product j over the selling horizon, we can view the linear program in (4) as an approximation to the network revenue management problem that is formulated under the assumption that the numbers of requests for the products take on their expected values. It is well-known that the optimal objective value of the linear program in (4) provides an upper bound on the optimal total expected revenue; see Bertsimas and Popescu (2003). In practice, this upper bound becomes useful when assessing the optimality gaps of various heuristics. In this section, we show that we can leverage an optimal solution to the linear program in (4) when constructing our value function approximations. We explain the idea using the model in Section 2, but we can incorporate customer choice behavior as shown in Section 4.1, and allow for products consuming multiple units of a resource as shown in Section 4.2. For a collection of availability-tracking basis functions $\mathscr{B} = \{\varphi_j : j \in \mathcal{J}\}$, we approximate the optimal value functions $\{V^t : t \in \mathcal{T}\}$ using value function approximations $\{H^t : t \in \mathcal{T}\}$ of the form $H^t(\boldsymbol{x}) = \sum_{j \in \mathcal{J}} \gamma_j^t \varphi_j(\boldsymbol{x})$. To compute the coefficients $\{\gamma_j^t : j \in \mathcal{J},\ t \in \mathcal{T}\}$, we use the following algorithm.

- **Initialization:** Let $\mathscr{B} = \{\varphi_j : j \in \mathcal{J}\}$ be any collection of availability-tracking basis functions, $\theta \geq \Delta_{\mathscr{B}}$ be a tuning parameter, and $(z_j^* : j \in \mathcal{J})$ be an optimal solution to the linear program in (4). Initialize $\gamma_j^{T+1} = 0$ for all $j \in \mathcal{J}$.
- **Coefficient Computation**: For each $t = T, T-1, \ldots, 1$, use the coefficients $\{\gamma_j^{t+1} : j \in \mathcal{J}\}$ to compute $\{\gamma_j^t : j \in \mathcal{J}\}$ as

$$\gamma_j^t \;=\; \frac{z_j^*}{\Lambda_j}\, \lambda_j^t \left[r_j - \theta \sum_{i \in A^j} \frac{1}{C_i} \sum_{k \in \mathcal{J}} \mathbb{1}_{\{i \in A^k\}}\, \gamma_k^{t+1} \right]^+ +\; \gamma_j^{t+1}. \tag{16}$$

Once we construct the value function approximations $\{H^t : t \in \mathcal{T}\}$ using the algorithm above, we use the same decision function in (5) in our approximate policy. The following theorem gives a performance guarantee for our approximate policy. The proof is in Appendix D.

Theorem 4.4 (Performance with Linear Programming Approximation) *If the tuning parameter θ satisfies $\theta \geq \Delta_{\mathscr{B}}$, then the total expected revenue obtained by the approximate policy is at least $1/(1+\theta L)$ fraction of the optimal.*

Intuitively, if product j is unlikely to contribute to the optimal total expected revenue, then we expect z_j^* to be close to zero. In this case, noting (16), the coefficients $\{\gamma_j^t : t \in \mathcal{T}\}$ for this product do not contribute significantly to the value function approximation.

5. Computational Experiments

In this section, we discuss computational experiments that we conducted to assess the numerical performance of our approximate policy. During the course of our computational experiments, we also experiment with different availability-tracking basis functions.

5.1 Experimental Setup

In our computational experiments, we use the test problems in Topaloglu (2009). A number of other papers, including Hu et al. (2013), Brown and Smith (2014), Vossen and Zhang (2015a,b), and Kunnumkal and Talluri (2016a), used these test problems in their computational experiments as well. The test problems in Topaloglu (2009) originate from the airline setting, where the resources correspond to the flight legs and the products correspond to the itineraries. In our test problems, the airline network has $N+1$ locations. One location is the hub and the remaining N locations are the spokes. There is a flight leg from each spoke to the hub and a flight leg from the hub to each spoke. Therefore, the number of flight legs is $2N$. In Figure 1, we show the structure of the airline network with $N=6$. We vary N in our computational experiments. Note that there are N origin-destination pairs that connect the hub to a spoke, N origin-destination pairs that connect a spoke to the hub, and $N(N-1)$ origin-destination pairs that connect a spoke to another spoke, resulting in $2N + N(N-1)$ origin-destination pairs. There is a high-fare and a low-fare itinerary that connects each origin-destination pair. Thus, the number of itineraries is $2\,(2N + N(N-1))$. For a certain origin-destination pair, the revenue associated with the high-fare itinerary connecting this origin-destination pair is κ times the revenue associated with the corresponding low-fare itinerary, where κ captures the revenue difference between the high-fare and low-fare itineraries. We vary κ in our computational experiments as well.

The arrival probabilities for the itinerary requests are generated in such a way that the requests for the high-fare itineraries tend to arrive later in the selling horizon. To generate the arrival probabilities $\{\lambda_j^t : j \in \mathcal{J},\ t \in \mathcal{T}\}$ for the itinerary requests, for each origin-destination pair (o,d),

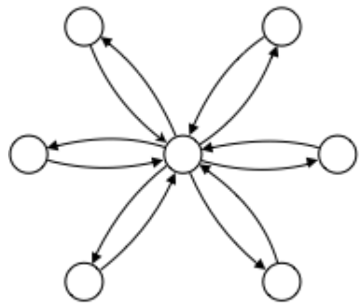

Figure 1 Structure of the airline network with $N=6$ spokes.

we sample $B_{o,d}$ from the uniform distribution over $[0,1]$ and $\tau_{o,d}$ from the uniform distribution over $\{\frac{1}{2}T,\ldots,\frac{2}{3}T\}$. In this case, the probability that we have a request for an itinerary that connects the origin-destination pair (o,d) is proportional to $B_{o,d}$. The probability that we have a request for the low-fare itinerary that connects the origin-destination pair (o,d) decreases over time, whereas the probability that we have a request for the corresponding high-fare itinerary is zero until time period $\tau_{o,d}$, but this probability increases over time after time period $\tau_{o,d}$. Therefore, the requests for the high-fare itinerary start appearing only after time period $\tau_{o,d}$. To be precise, after generating $B_{o,d}$ and $\tau_{o,d}$, for all $t\in\mathcal{T}$, we set $\eta^{\mathsf{t,low}}_{o,d}$ and $\eta^{\mathsf{t,high}}_{o,d}$ as

$$\eta^{\mathsf{t,low}}_{o,d}=B_{o,d}\,\frac{T+1-t}{T} \quad \text{and} \quad \eta^{\mathsf{t,high}}_{o,d}=\begin{cases}0 & \text{if } t\le\tau_{o,d}\\ B_{o,d}\,\frac{t-\tau_{o,d}}{T-\tau_{o,d}} & \text{otherwise.}\end{cases}$$

Letting $\mathcal{D}$ be the set of all origin-destination pairs, if itinerary j is the low-fare itinerary for origin-destination pair (o,d), then we set $\lambda^t_j=\eta^{\mathsf{t,low}}_{o,d}/\sum_{(s,r)\in\mathcal{D}}(\eta^{\mathsf{t,low}}_{s,r}+\eta^{\mathsf{t,high}}_{s,r})$, but if itinerary j is the corresponding high-fare itinerary, then we set $\lambda^t_j=\eta^{\mathsf{t,high}}_{o,d}/\sum_{(s,r)\in\mathcal{D}}(\eta^{\mathsf{t,low}}_{s,r}+\eta^{\mathsf{t,high}}_{s,r})$. Since the requests for the high-fare itineraries tend to arrive later in the selling horizon, it becomes important to reserve the capacities for the high-fare itinerary requests by rejecting the requests early in the selling horizon. The total expected demand for the capacity on flight leg i is $\sum_{t\in\mathcal{T}}\sum_{j\in\mathcal{J}}\mathbb{1}_{\{i\in A^j\}}\,\lambda^t_j$, so the initial capacity of flight leg i is set to be $C_i=\frac{1}{\alpha}\sum_{t\in\mathcal{T}}\sum_{j\in\mathcal{J}}\mathbb{1}_{\{i\in A^j\}}\,\lambda^t_j$. Thus, larger values for α yield tighter capacities. We vary α in our computational experiments.

Letting N, κ, and α be as above, and recalling that T is the length of the selling horizon, we vary $T\in\{200,600\}$, $N\in\{4,5,6,8\}$, $\kappa\in\{2,4\}$, and $\alpha\in\{1.0,1.2,1.6\}$, to get 48 test problems.

5.2 Benchmark Methods

In our computational experiments, we work with six benchmarks, one of which is our approximate policy. We proceed to describing our benchmarks.

Approximate Policy (APP): This benchmark is our approximate policy with the decision function in (5). We experimented with different basis functions. The basis function $\varphi_j(\boldsymbol{x})=$

$\min_{i \in A^j} (1 - e^{-x_i/C_i})/(1 - e^{-1})$, which is availability-tracking by the discussion in Section 3.1, provided consistent improvements over the others that we experimented with. Thus, we use this basis function. One can check that $\Delta_{\mathscr{B}} = 1/(1 - e^{-1})$ for this basis function. In our computational results, we discuss our experimentation with different basis functions. Note that our basis function resembles the tradeoff function $\Psi(x) = (1 - e^{-x})/(1 - e^{-1})$ in Golrezaei et al. (2014).

In our practical implementation of APP, we make two modifications. First, we divide the selling horizon into five equal segments and reconstruct our value function approximations at the beginning of each segment. In particular, the beginning of segment k corresponds to time period $(k-1)\frac{T}{5} + 1$. If the remaining capacities on the flight legs at the beginning of segment k are given by the vector $\boldsymbol{x}$, then we replace C_i in the recursion in (3) with x_i and use this recursion over time periods $T, T-1, \ldots, (k-1)\frac{T}{5} + 1$ to compute the coefficients $\{\gamma_j^t : j \in \mathcal{J},\ t = (k-1)\frac{T}{5} + 1, \ldots, T\}$. These coefficients specify the value function approximations that we use when making the decisions over segment k. When we reach the beginning of the next segment, we reconstruct our value function approximations in a similar fashion. Second, we calibrate the value for the tuning parameter θ at the beginning of each segment. The values of the coefficients $\{\gamma_j^t : j \in \mathcal{J},\ t \in \mathcal{T}\}$ in (3) depend on θ, which, in turn, implies that the total expected revenue obtained by APP also depends on θ. When reconstructing our value function approximations at the beginning of each segment, we search for the best tuning parameter over the interval $\mathcal{I} = [1/(1 - e^{-1}), 15]$ with a precision of 0.01. Given that we use the tuning parameter θ when reconstructing our value function approximations at the beginning of segment k, let $U^{k,\theta}(\boldsymbol{x})$ be the total expected revenue obtained by APP over time periods $(k-1)\frac{T}{5} + 1, \ldots, T$ starting with the capacities $\boldsymbol{x}$ for the flight legs. Computing the total expected revenue $U^{k,\theta}(\boldsymbol{x})$ exactly is intractable, because computing this quantity requires solving a dynamic program similar to the one in (8), but we estimate this quantity using simulation. At the beginning of segment k, we choose the value of the tuning parameter θ as $\arg\max\{U^{k,\theta}(\boldsymbol{x}) : \theta \in \mathcal{I} \cap \{0.01 \times \ell : \ell = 0, 1, \ldots\}\}$. We use this value for the tuning parameter until we reach the beginning of the next segment. The theoretical performance guarantee that we give in Theorem 3.5 will be the strongest when we use the smallest possible value for θ with $\theta \geq \Delta_{\mathscr{B}}$, but choosing a different value for the tuning parameter may actually provide better practical performance for APP.

Bid Price Policy (BPP): This benchmark is the well-known bid price policy; see Section 3.3 in Talluri and van Ryzin (2005). The idea behind BPP is to use the optimal values of the dual variables associated with the first constraint in the linear program in (4) to estimate the value of a unit of capacity on each flight leg. In this case, if the revenue from a certain itinerary exceeds the value of the capacities used by this itinerary, then we accept the request for the itinerary. To be specific, letting $(\mu_i^* : i \in \mathcal{L})$ be the optimal values of the dual variables associated with the first constraint in problem (4), BPP accepts a request for itinerary j if and only if $r_j \geq \sum_{i \in A^j} \mu_i^*$ and there are sufficient capacities to serve a request for itinerary j. In our practical implementation of BPP, we divide the selling horizon into five equal segments and resolve problem (4) at the beginning of each segment. In particular, if the remaining capacities on the flight legs at the beginning of segment k are given by the vector $\boldsymbol{x}$, then we replace C_i with x_i and Λ_j with $\sum_{t=(k-1)\,\frac{T}{5}+1}^{T} \lambda_j^t$ in problem (4). Letting $(\mu_i^* : i \in \mathcal{L})$ be the optimal values of the dual variables associated with the first constraint, we use these values of the dual variables during segment k. Using a similar approach, we recompute the policy parameters for all other benchmarks at the beginning of each segment, but for economy of space, we do not discuss recomputation any longer.

Randomized Linear Program (RLP): In this benchmark, we use the realizations of the total numbers of itinerary requests over the selling horizon, as opposed to their expected values, to capture the distribution information for the total numbers of itinerary requests. Using the random variable D_j to denote the total number of requests for itinerary j over the selling horizon, we replace Λ_j on the right side of the second constraint in problem (4) with D_j. As a function of $\boldsymbol{D} = (D_j : j \in \mathcal{J})$, letting $(\mu_i^*(\boldsymbol{D}) : i \in \mathcal{L})$ be the optimal values of the dual variables associated with the first constraint in problem (4), we use the $\mathbb{E}\{\mu_i^*(\boldsymbol{D})\}$ to estimate the value of a unit of capacity on flight leg i. In this case, RLP accepts a request for itinerary j as long as $r_j \geq \sum_{i \in A^j} \mathbb{E}\{\mu_i^*(\boldsymbol{D})\}$; see Talluri and van Ryzin (1999). Computing the expectation $\mathbb{E}\{\mu_i^*(\boldsymbol{D})\}$ exactly is intractable, so we estimate this expectation using simulation.

Finite Differences (DIF): Here, we use the optimal objective value of problem (4) to estimate the value of the capacities used by an itinerary. As a function of $\boldsymbol{C} = (C_i : i \in \mathcal{L})$, we let $\mathsf{Z}_{LP}^*(\boldsymbol{C})$ be the optimal objective value of problem (4), in which case, we estimate the value of the capacities used by itinerary j as $\mathsf{Z}_{LP}^*(\boldsymbol{C}) - \mathsf{Z}_{LP}^*(\boldsymbol{C} - \sum_{i \in A^j} \boldsymbol{e}_i)$. Thus, DIF accepts a request for itinerary j as long as $r_j \geq \mathsf{Z}_{LP}^*(\boldsymbol{C}) - \mathsf{Z}_{LP}^*(\boldsymbol{C} - \sum_{i \in A^j} \boldsymbol{e}_i)$; see Bertsimas and Popescu (2003).

Dynamic Programming Decomposition (DEC): The idea behind this benchmark is to decompose the dynamic programming formulation of the problem by the flight legs, in which case, we can solve dynamic programs with scalar state variables to obtain value function approximations; see Section 4.2 in Zhang and Adelman (2009). To our knowledge, DEC is one of the strongest heuristics in practice, but it does not have a performance guarantee.

Online Packing Policy (OPP): This benchmark uses a linear program similar to the one in (4) to construct a policy for online packing problems; see Kesselheim et al. (2014). If the arrivals are stationary, then this policy has the competitive ratio of $1 - O(\sqrt{(\log L)/c_{\min}})$ that we discuss in the introduction section, but in Appendix E, we give a simple example to show that this policy can perform arbitrarily poorly under nonstationary arrivals.

Params. (T, N, κ, α)	Upp. Bnd.	Tot. Exp. Rev. APP	BPP	RLP	DIF	DEC	OPP	% Gap with APP BPP	RLP	DIF	DEC	OPP
$(200, 4, 4, 1.0)$	21,531	20,013	19,377	19,937	19,608	20,076	19,601	3.18	0.38	2.02	-0.32	2.06
$(200, 4, 4, 1.2)$	19,882	18,386	17,140	17,964	17,529	18,538	17,276	6.78	2.29	4.66	-0.82	6.04
$(200, 4, 4, 1.6)$	17,530	15,993	14,474	15,712	14,996	16,185	13,507	9.50	1.76	6.24	-1.20	15.55
$(200, 4, 8, 1.0)$	34,571	32,655	30,692	32,447	31,305	32,845	31,858	6.01	0.64	4.14	-0.58	2.44
$(200, 4, 8, 1.2)$	32,922	31,020	27,324	30,310	28,384	31,284	28,025	11.91	2.29	8.50	-0.85	9.65
$(200, 4, 8, 1.6)$	30,570	28,704	24,062	27,890	25,461	28,861	21,901	16.17	2.83	11.30	-0.55	23.70
$(200, 5, 4, 1.0)$	22,144	20,984	20,197	20,796	20,488	21,139	20,478	3.75	0.90	2.36	-0.74	2.41
$(200, 5, 4, 1.2)$	21,263	19,565	18,462	19,225	18,933	19,716	18,463	5.64	1.73	3.23	-0.77	5.63
$(200, 5, 4, 1.6)$	18,870	17,037	15,406	16,676	16,090	17,260	14,925	9.57	2.12	5.56	-1.31	12.39
$(200, 5, 8, 1.0)$	35,387	33,943	31,844	33,519	32,491	34,219	33,048	6.18	1.25	4.28	-0.81	2.63
$(200, 5, 8, 1.2)$	34,495	32,318	29,232	31,551	30,395	32,653	29,702	9.55	2.37	5.95	-1.04	8.10
$(200, 5, 8, 1.6)$	32,081	29,666	24,971	28,943	26,800	30,068	24,074	15.82	2.44	9.66	-1.36	18.85
$(200, 6, 4, 1.0)$	22,300	20,595	19,819	20,496	20,110	20,699	19,943	3.77	0.48	2.35	-0.51	3.17
$(200, 6, 4, 1.2)$	20,932	19,049	17,927	18,753	18,463	19,174	17,793	5.89	1.55	3.08	-0.65	6.60
$(200, 6, 4, 1.6)$	18,592	16,595	15,220	16,302	15,863	16,786	14,235	8.29	1.76	4.41	-1.15	14.22
$(200, 6, 8, 1.0)$	35,544	33,338	31,132	33,205	31,946	33,644	32,186	6.62	0.40	4.17	-0.92	3.46
$(200, 6, 8, 1.2)$	34,172	31,623	28,504	31,107	29,737	32,004	28,603	9.86	1.63	5.96	-1.21	9.55
$(200, 6, 8, 1.6)$	31,824	29,191	24,923	28,459	26,702	29,551	22,880	14.62	2.51	8.53	-1.23	21.62
$(200, 8, 4, 1.0)$	20,052	18,359	17,508	17,875	17,742	18,421	17,634	4.63	2.63	3.36	-0.34	3.95
$(200, 8, 4, 1.2)$	18,952	16,936	15,753	16,354	16,188	17,054	15,728	6.98	3.43	4.41	-0.70	7.13
$(200, 8, 4, 1.6)$	16,833	14,676	13,371	14,161	14,019	14,831	12,439	8.89	3.51	4.48	-1.05	15.24
$(200, 8, 8, 1.0)$	31,835	29,742	27,378	28,779	28,058	29,890	28,346	7.95	3.24	5.66	-0.50	4.69
$(200, 8, 8, 1.2)$	30,727	28,232	24,793	27,116	25,956	28,412	25,218	12.18	3.95	8.06	-0.64	10.68
$(200, 8, 8, 1.6)$	28,608	25,913	21,844	24,837	23,616	26,116	19,869	15.70	4.15	8.87	-0.78	23.32
Average								8.73	2.09	5.47	-0.83	9.71

Table 1 Computational results for the test problems with $T = 200$ time periods in the selling horizon.

5.3 Computational Results

Table 1 shows our computational results on the test problems with $T = 200$ time periods in the selling horizon, while Table 2 shows our computational results on the test problems with $T = 600$. The layouts of the two tables are identical. In the first column, we show the parameter configuration for each test problem using the tuple (T, N, κ, α), where N, κ and α are as discussed in our experimental setup. In the second column, we show the upper bound on the optimal total expected revenue provided by the optimal objective value of problem (4). In the third to eighth columns, we show the total expected revenues obtained by APP, BPP, RLP, DIF, DEC and OPP. We estimate these total expected revenues by simulating the performance of each benchmark over 100 sample paths. In the ninth to thirteenth columns, we give the percent gaps between the total expected revenues obtained by APP and the remaining five benchmarks.

Considering the results in Table 1, the performance of APP is better than that of BPP with a substantial margin. RLP and DIF both perform better than BPP, but this improvement is not enough to catch up with APP. We underline two trends by comparing the performance of APP with that of RLP, but similar observations hold when we compare the performance of APP with that of BPP or DIF. We focus on comparing APP with RLP, because RLP already performs noticeably

Params. (T, N, κ, α)	Upp. Bnd.	Tot. Exp. Rev. APP	BPP	RLP	DIF	DEC	OPP	% Gap with APP BPP	RLP	DIF	DEC	OPP
$(600, 4, 4, 1.0)$	32,409	30,596	29,751	30,209	29,969	30,724	29,988	2.76	1.27	2.05	-0.42	1.99
$(600, 4, 4, 1.2)$	29,852	27,990	26,100	27,455	26,546	28,233	26,138	6.75	1.91	5.16	-0.87	6.62
$(600, 4, 4, 1.6)$	26,324	24,414	22,009	24,060	22,567	24,668	20,474	9.85	1.45	7.57	-1.04	16.14
$(600, 4, 8, 1.0)$	52,086	49,909	47,249	49,168	47,796	50,194	48,553	5.33	1.49	4.23	-0.57	2.72
$(600, 4, 8, 1.2)$	49,529	47,214	41,751	46,069	43,748	47,677	42,223	11.57	2.43	7.34	-0.98	10.57
$(600, 4, 8, 1.6)$	46,001	43,637	36,632	42,941	38,256	44,008	33,108	16.05	1.59	12.33	-0.85	24.13
$(600, 5, 4, 1.0)$	33,299	31,928	30,763	31,592	31,027	32,095	31,217	3.65	1.05	2.82	-0.52	2.23
$(600, 5, 4, 1.2)$	31,943	29,813	28,290	29,473	28,832	30,147	28,206	5.11	1.14	3.29	-1.12	5.39
$(600, 5, 4, 1.6)$	28,343	25,903	23,847	25,713	24,582	26,405	22,724	7.94	0.73	5.10	-1.94	12.27
$(600, 5, 8, 1.0)$	53,285	51,563	48,653	50,871	49,263	51,923	50,250	5.64	1.34	4.46	-0.70	2.55
$(600, 5, 8, 1.2)$	51,904	49,207	44,958	48,639	46,078	49,737	45,280	8.64	1.15	6.36	-1.08	7.98
$(600, 5, 8, 1.6)$	48,283	45,221	38,966	44,582	40,784	45,911	36,476	13.83	1.41	9.81	-1.52	19.34
$(600, 6, 4, 1.0)$	26,873	25,369	24,405	24,777	24,576	25,445	24,661	3.80	2.33	3.13	-0.30	2.79
$(600, 6, 4, 1.2)$	25,184	23,325	21,945	22,553	22,455	23,517	21,841	5.92	3.31	3.73	-0.82	6.36
$(600, 6, 4, 1.6)$	22,274	20,327	18,542	19,622	19,259	20,571	17,466	8.78	3.47	5.25	-1.20	14.07
$(600, 6, 8, 1.0)$	42,865	41,102	38,411	39,830	38,935	41,305	39,768	6.55	3.09	5.27	-0.49	3.24
$(600, 6, 8, 1.2)$	41,166	38,936	34,848	37,267	36,205	39,238	35,128	10.50	4.29	7.02	-0.77	9.78
$(600, 6, 8, 1.6)$	38,252	35,845	30,536	34,430	32,492	36,238	28,181	14.81	3.95	9.35	-1.09	21.38
$(600, 8, 4, 1.0)$	24,167	22,332	21,241	21,601	21,616	22,466	21,546	4.89	3.27	3.20	-0.60	3.52
$(600, 8, 4, 1.2)$	22,755	20,539	19,074	19,726	19,668	20,710	19,077	7.13	3.96	4.24	-0.83	7.12
$(600, 8, 4, 1.6)$	20,228	17,852	16,411	17,221	17,083	18,076	15,002	8.07	3.53	4.31	-1.25	15.97
$(600, 8, 8, 1.0)$	38,395	36,299	33,270	34,870	34,159	36,521	34,599	8.34	3.94	5.89	-0.61	4.68
$(600, 8, 8, 1.2)$	36,976	34,376	30,063	32,791	31,831	34,614	30,603	12.55	4.61	7.40	-0.69	10.97
$(600, 8, 8, 1.6)$	34,449	31,589	26,876	30,209	28,856	31,840	23,875	14.92	4.37	8.65	-0.79	24.42
Average								8.47	2.55	5.75	-0.88	9.84

Table 2 Computational results for the test problems with $T = 600$ time periods in the selling horizon.

better than BPP and DIFF, which are the two other benchmarks that are based on the linear program in (4). The first trend is that as the parameter α increases and the capacities on the flight legs get tighter, the performance gaps between APP and RLP get larger. Considering the test problems with $\alpha = 1.0$, $\alpha = 1.2$, and $\alpha = 1.6$ separately, the average percent gaps between the total expected revenues obtained by APP and RLP are, respectively, 1.24%, 2.41%, and 2.64%. As the capacities on the flight legs get tighter, it becomes more important to protect the capacity for the high-fare itinerary requests that tend to arrive later in the selling horizon. It appears that APP does a better job of capturing this tradeoff. The second trend is that as the parameter κ increases and the revenue difference between the high-fare and low-fare itineraries increases, the performance gap between APP and RLP increases as well. Considering the test problems with $\kappa = 2$ and $\kappa = 4$ separately, the average percent gaps between the total expected revenues obtained by APP and RLP are, respectively, 1.88% and 2.31%. When the revenue difference between the high-fare and low-fare itineraries increases, the opportunity cost of not having the capacity to serve a high-fare itinerary request also increases and APP seems to do a better job of reserving the capacity for high-fare itinerary requests. APP lags behind DEC with a small but consistent margin. Over all test problems, the average gap between the performance of APP and DEC is 0.83%. As mentioned earlier, to our knowledge, DEC is one of the strongest heuristics for network revenue management problems in practice, but we emphasize that DEC does not have a theoretical performance guarantee. Similar to the trends discussed earlier in this paragraph, the performance gap between APP and DEC gets larger as the capacities on the flight legs get tighter or the revenue difference between the high-fare and low-fare itineraries increases. The performance of OPP is not competitive to APP, which is not surprising because OPP is designed to deal with stationary arrivals. Similar observations hold for the results in Table 2. APP consistently performs better than BPP. Although RLP and DIF both perform better than BPP, they do not catch up with APP. APP lags behind DEC by a small but consistent margin. OPP significantly lags behind APP.

The calibrated value of the tuning parameter θ that we use for APP depends on the problem parameters. In Table 3, we show the value of the tuning parameter for all of our test problems when we calibrate the tuning parameter at the beginning of the selling horizon. Our results indicate that the calibrated value of the tuning parameter gets smaller as α increases and the capacities on the flight legs get tighter. Intuitively speaking, as the capacities on the flight legs get tighter and the resources become more scarce, we expect the value of a unit of capacity to increase. Indeed, we numerically observed that if we decrease the value of θ in the recursion in (3), then values of the coefficients $\{\gamma_j^t : j \in \mathcal{J}\ , t \in \mathcal{T}\}$ computed through this recursion tend to increase, in which case, the value of a unit of capacity also increases. We carried out all of our computational experiments using Java 1.8.0 on 2.8 GHz Intel Xeon E5-2680 CPU with 1 GB of RAM. For the largest test problems with $T = 600$ time periods and $N = 8$ spokes, the average CPU time to compute the coefficients $\{\gamma_j^t : j \in \mathcal{J},\ t \in \mathcal{T}\}$ for a fixed value of θ was about 0.05 seconds.

In our implementation of APP, we experimented with five other availability-tracking basis functions, which are $\varphi_j^{\textsf{prd-exp}}(\boldsymbol{x}) = \prod_{i \in A^j}(1 - e^{-x_i/C_i})/(1 - e^{-1})$, $\varphi_j^{\textsf{min}}(\boldsymbol{x}) = \min_{i \in A^j} \frac{x_i}{C_i}$, $\varphi_j^{\textsf{prd}}(\boldsymbol{x}) = \prod_{i \in A^j} \frac{x_i}{C_i}$, $\varphi_j^{\textsf{exp-sum}}(\boldsymbol{x}) = \exp\left(\sum_{i \in A^j}\left(1 - \frac{C_i}{x_i}\right)\right)$ and $\varphi_j^{\textsf{recip-sum}}(\boldsymbol{x}) = |A^j| / \sum_{i \in A^j} \frac{C_i}{x_i}$. In Table 4, for economy of space, we consider a subset of our test problems and show the performance of APP with the five basis functions above, along with $\varphi_j^{\textsf{min-exp}}(\boldsymbol{x}) = \min_{i \in A^j}(1 - e^{-x_i/C_i})/(1 - e^{-1})$ used in our earlier computational results. In the first column, we show the parameter configuration for each test problem. In the second to seventh columns, we show the total expected revenues obtained by APP with the six basis functions. In the eighth to twelfth columns, we show the percent gap between the performance of APP with the basis function $\varphi_j^{\textsf{min-exp}}(\boldsymbol{x}) = \min_{i \in A^j}(1 - e^{-x_i/C_i})/(1 - e^{-1})$ and the remaining five basis functions. Our results indicate that the performance of APP is somewhat robust to the choice of basis functions, but by experimenting with different basis functions, we can improve the performance by about 1.20% on average.

In all of our test problems, the maximum number of resources used by a product is two. The theoretical performance guarantee that we give for APP in Theorem 3.5 depends on the maximum

Params. (T, N, κ, α)	θ	Params. (T, N, κ, α)	θ	Params. (T, N, κ, α)	θ	Params. (T, N, κ, α)	θ
$(200, 4, 4, 1.0)$	1.91	$(200, 6, 4, 1.0)$	1.78	$(600, 4, 4, 1.0)$	1.82	$(600, 6, 4, 1.0)$	1.60
$(200, 4, 4, 1.2)$	1.60	$(200, 6, 4, 1.2)$	1.62	$(600, 4, 4, 1.2)$	1.62	$(600, 6, 4, 1.2)$	1.59
$(200, 4, 4, 1.6)$	1.59	$(200, 6, 4, 1.6)$	1.59	$(600, 4, 4, 1.6)$	1.59	$(600, 6, 4, 1.6)$	1.60
$(200, 4, 8, 1.0)$	5.50	$(200, 6, 8, 1.0)$	5.64	$(600, 4, 8, 1.0)$	5.66	$(600, 6, 8, 1.0)$	5.65
$(200, 4, 8, 1.2)$	4.75	$(200, 6, 8, 1.2)$	4.75	$(600, 4, 8, 1.2)$	4.86	$(600, 6, 8, 1.2)$	4.81
$(200, 4, 8, 1.6)$	3.76	$(200, 6, 8, 1.6)$	3.98	$(600, 4, 8, 1.6)$	3.69	$(600, 6, 8, 1.6)$	3.93
$(200, 5, 4, 1.0)$	2.23	$(200, 8, 4, 1.0)$	1.61	$(600, 5, 4, 1.0)$	1.97	$(600, 8, 4, 1.0)$	1.71
$(200, 5, 4, 1.2)$	1.65	$(200, 8, 4, 1.2)$	1.59	$(600, 5, 4, 1.2)$	1.60	$(600, 8, 4, 1.2)$	1.59
$(200, 5, 4, 1.6)$	1.59	$(200, 8, 4, 1.6)$	1.59	$(600, 5, 4, 1.6)$	1.60	$(600, 8, 4, 1.6)$	1.59
$(200, 5, 8, 1.0)$	6.33	$(200, 8, 8, 1.0)$	5.03	$(600, 5, 8, 1.0)$	6.20	$(600, 8, 8, 1.0)$	5.50
$(200, 5, 8, 1.2)$	4.84	$(200, 8, 8, 1.2)$	4.37	$(600, 5, 8, 1.2)$	4.84	$(600, 8, 8, 1.2)$	4.38
$(200, 5, 8, 1.6)$	3.99	$(200, 8, 8, 1.6)$	3.40	$(600, 5, 8, 1.6)$	3.93	$(600, 8, 8, 1.6)$	3.45

Table 3 Calibrated values of the tuning parameter θ.

	Tot. Exp. Rev.						% Gap with Min-Exp				
Params (T, N, κ, α)	Min-Exp	Prd-Exp	Min	Prd	Exp-Sum	Recip-Sum	Prd-Exp	Min	Prd	Exp-Sum	Recip-Sum
$(200, 6, 4, 1.0)$	20,595	20,420	20,447	20,086	20,284	20,427	0.85	0.72	2.47	1.51	0.82
$(200, 6, 4, 1.2)$	19,049	18,847	18,820	18,341	18,586	18,874	1.06	1.20	3.72	2.43	0.92
$(200, 6, 4, 1.6)$	16,595	16,621	16,459	16,149	16,337	16,539	-0.16	0.82	2.69	1.56	0.33
$(200, 6, 8, 1.0)$	33,338	33,248	33,123	32,544	32,914	33,088	0.27	0.64	2.38	1.27	0.75
$(200, 6, 8, 1.2)$	31,623	31,619	31,425	30,816	31,059	31,192	0.01	0.63	2.55	1.78	1.36
$(200, 6, 8, 1.6)$	29,191	29,265	28,928	28,654	29,046	29,090	-0.25	0.90	1.84	0.50	0.34

Table 4 Performance of APP with different basis functions.

number of resources used by a product. In Appendix F, we provide computational experiments in the hotel network revenue management setting, where we systematically vary the maximum number of resources used by a product. Our results demonstrate that APP maintains its edge over BPP, RLP, DIF and OPP, and the performance gap between APP and DEC remains stable. Lastly, in our computational experiments, the arrivals for the product requests are nonstationary. OPP has a competitive ratio under stationary arrivals. In Appendix G, we provide computational experiments to test the performance of OPP under stationary arrivals. Under stationary arrivals, the performance of OPP gets better, but APP still provides significant improvements over OPP. An examination of Theorem 3 and Lemma 7 in Kesselheim et al. (2014) shows that the competitive ratio of OPP is $\max\{1 - 45\sqrt{(1 + \log_2 L)/c_{\min}},\, 1/(8\,e\,(2L)^{1/(c_{\min}-1)})\}$ for $c_{\min} \geq 2$, which, although approaches one as the capacities of the resource get large, can be substantially less than one for practical instances. For example, for $c_{\min} = 100$ and $L = 2$, this competitive ratio is about 0.045, which is significantly less than the performance guarantee of $1/(1 + L) = 1/3$ for APP. Note that OPP does not use forecasts of the numbers of requests for different products, which can partly explain its poor performance. The practical performance of OPP is not competitive, but it is remarkable that OPP has an asymptotic performance guarantee without using forecasts.

In all of our computational experiments, we recompute the policy parameters five times over the selling horizon. In Appendix H, we test the performance of the benchmarks when we compute the policy parameters only once at the beginning of the selling horizon. Lastly, we also implemented the approach in Section 4.3, which uses an optimal solution to problem (4) to compute the coefficients $\{\gamma_j^t : j \in \mathcal{J},\ t \in \mathcal{T}\}$. For our test problems, this approach did not provide noticeable improvements in the performance of APP.

6. Conclusion

We developed an approximate policy with a performance guarantee for network revenue management problems, which, to our knowledge, is unique as it works under nonstationary arrivals. In the paper, we pointed out several extensions of our approximate policy, but there are still other extensions that are possible. For example, in some papers in the online packing literature, there are multiple service modes with different revenues and resource consumptions. If we choose to accept a product request, then we decide which mode to use to serve the request; see, for example, Feldman et al. (2010) and Kesselheim et al. (2014). We can incorporate multiple service modes into our approximate policy by using our extension to the customer choice behavior given in Section 4.1. In particular, we use $\mathcal{K}$ to denote the set of service modes. At time period t, we have a request for product j with probability λ_j^t. If we use mode k to serve a request for product j, then we generate a revenue of $r_{j,k}$ and consume one unit of capacity for each resource in the set $A^{j,k} \subseteq \mathcal{L}$. We can reformulate the problem with multiple service modes equivalently as an instance of the problem in Section 4.1. In our reformulation, we refer to each product and service mode combination as a meta-product. At each time period, we choose a subset of meta-products to offer to the customers. Offering meta-product $(j,k) \in \mathcal{J} \times \mathcal{K}$ corresponds to being willing to use mode k to serve a request for product j. Not offering any of the meta-products $\{(j,k) : k \in \mathcal{K}\}$ corresponds to not being willing to accept a request for product j. The feasible subsets of meta-products that we can offer to the customers at a particular time period is given by $\mathcal{F} = \{S \subseteq \mathcal{J} \times \mathcal{K} : |S \cap \{(j,k) : k \in \mathcal{K}\}| \leq 1\ \forall j \in \mathcal{J}\}$, meaning that for each product j, we can offer at most one meta-product of the form (j,k). In this way, we ensure that if we are willing to accept a request for product j, then we choose one mode to serve it. If we offer the subset S of meta-products at time period t such that $(j,k) \in S$ for service mode k and we have a request for product j, then the arriving customer chooses meta-product (j,k). Therefore, we have the choice probability $\phi_{j,k}^t(S) = \lambda_j^t$ if $(j,k) \in S$. Otherwise, $\phi_{j,k}^t(S) = 0$. The revenue of meta-product (j,k) is $r_{j,k}$. If we sell one unit of meta-product (j,k), then we consume the capacities of the resources in the set $A^{j,k}$. Thus, replacing the products in the formulation in Section 4.1 with the meta-products, we can check that the feasible subsets of meta-products and the choice probabilities given above satisfy Assumption 4.1. So, we can use the formulation in Section 4.1 to come up with an approximate policy to decide which meta-products to offer at each time period to maximize the total expected revenue, which, in turn, yields an approximate policy to decide which product requests to accept and which mode to use for the accepted requests.

By the preceding discussion, the extension in Section 4.1 allows us to incorporate multiple service modes, but there are other variants of the network revenue management problem that are difficult to address by using our approach. For example, if overbooking is allowed, then the dynamic programming formulation of the problem is fundamentally different than the one that we use, because the state variable needs to keep track of the number of accepted bookings for each product, which ultimately may or may not show up. Thus, extending our work to handle overbooking is not straightforward. Also, the choice of our basis functions and the algorithm that we use to construct the coefficients of the basis functions strictly exploit the structure of the network revenue management problem. Extending our approach to a broader class of dynamic programs is certainly worthwhile, but such extensions appear to be nontrivial to us at this point. Another important point is that our choice of the basis functions was based on experimentation. The definition of availability-tracking basis functions provides some guidance on the choice of the basis functions, but a more systematic approach for choosing the basis functions is a useful research direction. Moreover, although our approach has a performance guarantee, this performance guarantee stays away from one. It would be useful if we can establish that our approach becomes asymptotically optimal in some regime, such as the one where the resource capacities and the expected demands for the products increase linearly with the same rate. Lastly, the dynamic programming decomposition approach, which we used as a benchmark in our computational experiments, is one of the strongest heuristics for network revenue management problems. To our knowledge, however, this approach does not have a performance guarantee. It would be useful to understand whether it is possible to give a performance guarantee for this approach.

Acknowledgements. We gratefully acknowledge the comments of the area editor, associate editor and three referees, which substantially improved our exposition, technical results and computational experiments. The second and fourth authors were supported in part by National Science Foundation grants CMMI-1824860 and CMMI-1825406. The fourth author was also supported in part by Schmidt Sciences.

Author Biographies: Yuhang Ma is a PhD student in the School of Operations Research and Information Engineering at Cornell University. Her research interests are assortment optimization and revenue management.

Paat Rusmevichientong is a professor of data sciences and operations at the USC Marshall School of Business. His research interests include revenue management, choice modeling, pricing, assortment optimization, and approximate dynamic programming.

Mika Sumida is a PhD student in the School of Operations Research and Information Engineering at Cornell University. Her research interests include algorithm design, resource allocation, and revenue management with applications in online marketplaces and the sharing economy.

Huseyin Topaloglu is a professor in the School of Operations Research and Information Engineering at Cornell University. His research focuses on revenue management, assortment optimization, transportation logistics, and approximate dynamic programming.

References

Adelman, D. 2007. Dynamic bid prices in revenue management. *Operations Research* **55**(4) 647–661.

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

Asadpour, A., H. Nazerzadeh. 2016. Maximizing stochastic monotone submodular functions. *Management Science* **62**(8) 2374–2391.

Bertsimas, D., I. Popescu. 2003. Revenue management in a dynamic network environment. *Transportation Science* **37**(3) 257–277.

Bront, J. J. M., I. Mendez Diaz, G. Vulcano. 2009. A column generation algorithm for choice-based network revenue management. *Operations Research* **57**(3) 769–784.

Brown, D. B., J. E. Smith. 2014. Information relaxations, duality, and convex stochastic dynamic programs. *Operations Research* **62**(6) 1394–1415.

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

Chan, C. W., V. F. Farias. 2009. Stochastic depletion problems: Effective myopic policies for a class of dynamic optimization problems. *Mathematics of Operations Research* **34**(2) 333–350.

Chaneton, J. M., G. Vulcano. 2011. Computing bid prices for revenue management under customer choice behavior. *Manufacturing & Service Operations Management* **13**(4) 452–470.

Cooper, W. L. 2002. Asymptotic behavior of an allocation policy for revenue management. *Operations Research* **50**(4) 720–727.

Cooper, W. L., T. Homem de Mello. 2007. Some decomposition methods for revenue management. *Transportation Science* **41**(3) 332–353.

Devanur, N. R., T. P. Hayes. 2009. The adwords problem: Online keyword matching with budgeted bidders under random permutations. *Proceedings of the ACM Conference on Electronic Commerce*. 71–78.

Devanur, N. R., K. Jain, B. Sivan, C. A. Wilkens. 2011. Near optimal online algorithms and fast approximation algorithms for resource allocation problems. *Proceedings of the 12th ACM Conference on Electronic Commerce*. EC '11, ACM, New York, NY, USA, 29–38.

Feldman, J., M. Henzinger, N. Korula, V. S. Mirrokni, C. Stein. 2010. Online stochastic packing applied to display ad allocation. *Proceedings of the 18th Annual European Conference on Algorithms: Part I*. ESA'10, Springer-Verlag, Berlin, Heidelberg, 182–194.

Gallego, G., G. Iyengar, R. Phillips, A. Dubey. 2004. Managing flexible products on a network. CORC Technical Report TR-2004-01.

Gallego, G., A. Li, V. A. Truong, X. Wang. 2016. Online bipartite matching. Working Paper, Columbia University.

Goel, A., M. Mahdian, H. Nazerzadeh, A. Saberi. 2010. Advertisement allocation for generalized second-pricing schemes. *Operations Research Letters* **38**(6) 571–576.

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

Hu, X., R. Caldentey, G. Vulcano. 2013. Revenue sharing in airline alliances. *Management Science* **59**(5) 1177–1195.

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

Jasin, S., S. Kumar. 2013. Analysis of deterministic LP-based booking limit and bid price controls for revenue management. *Operations Research* **61**(6) 1312–1320.

Kesselheim, T., A. Tönnis, K. Radke, B. Vöcking. 2014. Primal beats dual on online packing LPs in the random-order model. *Proceedings of the Forty-sixth Annual ACM Symposium on Theory of Computing*. STOC '14, New York, NY, 303–312.

Kirshner, S. N., M. Nediak. 2015. Scalable dynamic bid prices for network revenue management in continuous time. *Production and Operations Management* **24**(10) 1621–1635.

Kunnumkal, S., K. Talluri. 2016a. On a piecewise-linear approximation for network revenue management. *Mathematics of Operations Research* **41**(1) 72–91.

Kunnumkal, S., K. Talluri. 2016b. Technical note: A note on relaxations of choice network revenue management dynamic program. *Operations Research* **64**(1) 158–166.

Kunnumkal, S., H. Topaloglu. 2008. A refined deterministic linear program for the network revenue management problem with customer choice behavior. *Naval Research Logistics Quarterly* **55**(6) 563–580.

Kunnumkal, S., H. Topaloglu. 2010a. Computing time-dependent bid prices in network revenue management problems. *Transportation Science* **44**(1) 38–62.

Kunnumkal, S., H. Topaloglu. 2010b. A new dynamic programming decomposition method for the network revenue management problem with customer choice behavior. *Production and Operations Management* **19**(5) 575–590.

Liu, Q., G. J. van Ryzin. 2008. On the choice-based linear programming model for network revenue management. *Manufacturing & Service Operations Management* **10**(2) 288–310.

Maglaras, C., J. Meissner. 2006. Dynamic pricing strategies for multiproduct revenue management problems. *Manufacturing & Service Operations Management* **8**(2) 136–148.

Mehta, A., Saberi A, U. Vazirani, V. Vazirani. 2007. Adwords and generalized online matching. *Journal of the ACM* **54**(5) 22:1–22:19.

Meissner, J., A. Strauss, K. Talluri. 2012. An enhanced concave program relaxation for choice network revenue management. *Production and Operations Management* **22**(1) 71–87.

Mendez-Diaz, I., J. J. M. Bront, G. Vulcano, P. Zabala. 2010. A branch-and-cut algorithm for the latent-class logit assortment problem. *Discrete Applied Mathematics* **36** 383–390.

Molinaro, M., R. Ravi. 2014. The geometry of online packing linear programs. *Mathematics of Operations Research* **39**(1) 46–59.

Rusmevichientong, P., M. Sumida, H. Topaloglu. 2017. Dynamic assortment optimization for reusable products with random usage durations. Working Paper, Cornell Tech.

Simpson, R. W. 1989. *Using network flow techniques to find shadow prices for market demands and seat inventory control*. MIT, Department of Aeronautics and Astronautics, Flight Transportation Laboratory.

Strauss, A. K., K. Talluri. 2017. Tractable consideration set structures for network revenue management. *Production and Operations Management* **26**(7) 1359–1368.

Talluri, K. 2014. New formulations for choice network revenue management. *INFORMS Journal on Computing* **26**(2) 401–413.

Talluri, K. T., G. J. van Ryzin. 1998. An analysis of bid-price controls for network revenue management. *Management Science* **44**(11-part-1) 1577–1593.

Talluri, K. T., G. J. van Ryzin. 1999. A randomized linear programming method for computing network bid prices. *Transportation Science* **33**(2) 207–216.

Talluri, K. T., G. J. van Ryzin. 2005. *The theory and practice of revenue management*. Kluwer Academic Publishers, Boston, MA.

Tan, B., R. Srikant. 2012. Online advertisement, optimization and stochastic networks. *IEEE Transactions on Automatic Control* **57**(11) 2854–2868.

Tong, C., H. Topaloglu. 2013. On the approximate linear programming approach for network revenue management problems. *INFORMS Journal on Computing* **26**(1) 121–134.

Topaloglu, H. 2008. A stochastic approximation method to compute bid prices in network revenue management problems. *INFORMS Journal on Computing* **20**(4) 596–610.

Topaloglu, H. 2009. Using Lagrangian relaxation to compute capacity-dependent bid prices in network revenue management. *Operations Research* **57**(3) 637–649.

van Ryzin, G., G. Vulcano. 2008a. Computing virtual nesting controls for network revenue management under customer choice behavior. *Manufacturing & Service Operations Management* **10**(3) 448–467.

van Ryzin, G., G. Vulcano. 2008b. Simulation-based optimization of virtual nesting controls for network revenue management. *Operations Research* **56**(4) 865–880.

Vossen, T. W. M., D. Zhang. 2015a. A dynamic disaggregation approach to approximate linear programs for network revenue management. *Production and Operations Management* **24**(3) 469–487.

Vossen, T. W. M., D. Zhang. 2015b. Reductions of approximate linear programs for network revenue management. *Operations Research* **63**(6) 1352–1371.

Wang, X., V. A. Truong, D. Bank. 2016. Online advance admission scheduling for services, with customer preferences. Working Paper, Columbia University.

Williamson, E. L. 1992. Airline network seat control. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA.

Zhang, D. 2011. An improved dynamic programming decomposition approach for network revenue management. *Manufacturing & Service Operations Management* **13**(1) 35–52.

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

Zhang, D., W. L. Cooper. 2005. Revenue management for parallel flights with customer choice behavior. *Operations Research* **53**(3) 415–431.

Zhang, D., W. L. Cooper. 2009. Pricing substitutable flights in airline revenue management. *European Journal of Operational Research* **197**(3) 848–861.

Electronic Companion:

An Approximation Algorithm for Network Revenue Management under Nonstationary Arrivals

Yuhang Ma, Paat Rusmevichientong, Mika Sumida, Huseyin Topaloglu

Appendix A: Constructing Availability-Tracking Basis Functions

In the next lemma, we show that we can construct availability-tracking basis functions from existing ones by taking a composition with a non-decreasing function. By using this lemma, we can use known availability-tracking basis functions to construct others, enriching the class of availability-tracking basis functions we have access to.

Lemma A.1 *Let $\mathscr{B} = \{\varphi_j : j \in \mathcal{J}\}$ be a collection of availability-tracking basis functions. If $f : [0,1] \to [0,1]$ is non-decreasing and differentiable with $f(0) = 0$, $f(1) = 1$ and $\max_{a\in[0,1]} f'(a) < \infty$, then letting $\rho_j(\boldsymbol{x}) = f(\varphi_j(\boldsymbol{x}))$ for all $j \in \mathcal{J}$, the collection $\mathscr{D} = \{\rho_j : j \in \mathcal{J}\}$ is also availability-tracking with $\Delta_{\mathscr{D}} \le \big(\max_{a\in[0,1]} f'(a)\big) \times \Delta_{\mathscr{B}}$.*

Proof: It is simple to verify that $\mathscr{D}$ satisfies the three parts of Definition 3.1. Since $\varphi_j(\boldsymbol{C}) = 1$ and $\varphi_j(\boldsymbol{C} - \boldsymbol{e}_i) \le 1$, we have $\Delta_{\mathscr{B}} = \max_{j\in\mathcal{J},\, i\in\mathcal{L}} \max_{\boldsymbol{x}\in\mathcal{Q}\,:\, x_i \ge 1} C_i \times (\, \varphi_j(\boldsymbol{x}) - \varphi_j(\boldsymbol{x} - \boldsymbol{e}_i)\,) \ge 0$. In this case, by the mean value theorem, for each $\boldsymbol{x} \in \mathcal{Q}$ with $x_i \ge 1$, there exists $y \in [0,1]$ such that

$$C_i \times \Big(f(\varphi_j(\boldsymbol{x})) - f(\varphi_j(\boldsymbol{x} - \boldsymbol{e}_i)) \Big) = C_i \times \Big(f'(y)\, (\varphi_j(\boldsymbol{x}) - \varphi_j(\boldsymbol{x} - \boldsymbol{e}_i)) \Big) \le f'(y)\, \Delta_{\mathscr{B}} \le \max_{a\in[0,1]} f'(a) \times \Delta_{\mathscr{B}},$$

where the first inequality uses the fact that $f'(y) \ge 0$ and the definition of $\Delta_{\mathscr{B}}$, whereas the second inequality uses the fact that $\Delta_{\mathscr{B}} \ge 0$. The inequality above implies that $\Delta_{\mathscr{D}} = \max_{j\in\mathcal{J},\, i\in\mathcal{L}} \max_{\boldsymbol{x}\in\mathcal{Q}\,:\, x_i\ge 1} C_i \times (\, f(\varphi_j(\boldsymbol{x})) - f(\varphi_j(\boldsymbol{x} - \boldsymbol{e}_i))\,) \le \big(\max_{a\in[0,1]} f'(a)\big) \times \Delta_{\mathscr{B}}$. $\square$

The next lemma also allows us to use known availability-tracking basis functions to construct other availability-tracking basis functions.

Lemma A.2 *Let $\mathscr{B} = \{\varphi_j : j \in \mathcal{J}\}$ be a collection of availability-tracking basis functions. If, for each $i \in \mathcal{L}$, $g_i : [0,1] \to [0,1]$ is non-decreasing and differentiable with $g_i(0) = 0$, $g_i(1) = 1$ and $\max_{a\in[0,1]} g'(a) < \infty$, then letting $\rho_j(\boldsymbol{x}) = \min_{i\in A^j} g_i(x_i/C_i)$ or $\rho_j(\boldsymbol{x}) = \prod_{i\in A^j} g_i(x_i/C_i)$ for all $j \in \mathcal{J}$, the collection $\mathscr{D} = \{\rho_j : j \in \mathcal{J}\}$ is also availability-tracking with $\Delta_{\mathscr{D}} \le \max_{i\in\mathcal{L}} \max_{a\in[0,1]} g_i'(a)$.*

Proof: It is simple to verify that $\mathscr{D}$ satisfies the three parts of Definition 3.1. Also, by the mean value theorem, for $x_i = 1, \ldots, C_i$, there exists $y \in [0,1]$ such that $g_i(\frac{x_i}{C_i}) - g_i(\frac{x_i - 1}{C_i}) = \frac{1}{C_i}\, g_i'(y) \le$

$\frac{1}{C_i}\max_{\ell\in\mathcal{L}}\max_{a\in[0,1]} g'_\ell(a)$. First, consider the case $\rho_j(\boldsymbol{x}) = \min_{i\in A^j} g_i(x_i/C_i)$. Using the same line of reasoning that we use in Example 3.3 and noting that g_i is non-decreasing, we obtain

$$\begin{aligned}
\rho_j(\boldsymbol{x}) - \rho_j(\boldsymbol{x}-\boldsymbol{e}_i) &= \min_{\ell\in A^j}\left\{ g_i\left(\tfrac{x_\ell}{C_\ell}\right)\right\} - \min_{\ell\in A^j}\left\{ g_\ell\left(\tfrac{x_\ell - \mathbb{1}_{\{\ell=i\}}}{C_\ell}\right)\right\} \\
&= \begin{cases} g_i\left(\frac{x_i}{C_i}\right) - g_i\left(\frac{x_i-1}{C_i}\right) & \text{if } \; g_i\left(\frac{x_i}{C_i}\right) = \min_{\ell\in A^j} g_\ell\left(\frac{x_\ell}{C_\ell}\right) \\ \min_{\ell\in A^j} g_\ell\left(\frac{x_\ell}{C_\ell}\right) - g_i\left(\frac{x_i-1}{C_i}\right) & \text{if } \; g_i\left(\frac{x_i}{C_i}\right) > \min_{\ell\in A^j} g_\ell\left(\frac{x_\ell}{C_\ell}\right) \ge g_i\left(\frac{x_i-1}{C_i}\right) \\ 0 & \text{if } \; g_i\left(\frac{x_i-1}{C_i}\right) > \min_{\ell\in A^j} g_\ell\left(\frac{x_\ell}{C_\ell}\right). \end{cases} \\
&\le g_i\left(\tfrac{x_i}{C_i}\right) - g_i\left(\tfrac{x_i-1}{C_i}\right) \;\le\; \frac{1}{C_i}\max_{\ell\in\mathcal{L}}\max_{a\in[0,1]} g'_\ell(a).
\end{aligned}$$

Second, consider the case $\rho_j(\boldsymbol{x}) = \prod_{i\in A^j} g_i(x_i/C_i)$. Using the same line of reasoning that we use in Example 3.4 and noting, once again, that g_i is non-decreasing, we get

$$\begin{aligned}
\rho_j(\boldsymbol{x}) - \rho_j(\boldsymbol{x}-\boldsymbol{e}_i) &= \prod_{\ell\in A^j} g_\ell\left(\tfrac{x_\ell}{C_\ell}\right) \;-\; g_i\left(\tfrac{x_i-1}{C_i}\right) \;\cdot \prod_{\ell\in A^j\setminus\{i\}} g_\ell\left(\tfrac{x_\ell}{C_\ell}\right) \\
&= \left(g_i\left(\tfrac{x_i}{C_i}\right) - g_i\left(\tfrac{x_i-1}{C_i}\right)\right)\cdot \prod_{\ell\in A^j\setminus\{i\}} g_\ell\left(\tfrac{x_\ell}{C_\ell}\right) \;\le\; g_i\left(\tfrac{x_i}{C_i}\right) - g_i\left(\tfrac{x_i-1}{C_i}\right) \;\le\; \frac{1}{C_i}\max_{\ell\in\mathcal{L}}\max_{a\in[0,1]} g'_\ell(a).
\end{aligned}$$

Thus, we have $\rho_j(\boldsymbol{x}) - \rho_j(\boldsymbol{x}-\boldsymbol{e}_i) \le \frac{1}{C_i}\max_{\ell\in\mathcal{L}}\max_{a\in[0,1]} g'_\ell(a)$ in both cases. The last inequality implies that $\Delta_{\mathscr{D}} \le \max_{\ell\in\mathcal{L}}\max_{a\in[0,1]} g'_\ell(a)$. $\square$

In the rest of this section, we show that if $\mathscr{B} = \{\varphi_j : j\in\mathcal{J}\}$ is any collection of availability-tracking basis functions, then we can construct a collection of *non-decreasing* availability-tracking basis functions $\mathscr{D} = \{\rho_j : j\in\mathcal{J}\}$ such that $\Delta_{\mathscr{D}} \le \Delta_{\mathscr{B}}$. In particular, given any collection of availability-tracking basis functions $\mathscr{B} = \{\varphi_j : j\in\mathcal{J}\}$, we define $\rho_j : \mathcal{Q}\to[0,1]$ as

$$\rho_j(\boldsymbol{x}) \quad = \max_{(\delta_i\,:\,i\in A^j)\in\mathbb{Z}_+^{|A^j|}} \left\{ \varphi_j\left(\boldsymbol{x} - \sum_{i\in A^j}\delta_i\,\boldsymbol{e}_i\right) \;:\; 0\le\delta_i\le x_i \quad \forall\, i\in A^j\right\}. \tag{17}$$

We will show three results for the collection of basis-functions $\mathscr{D} = \{\rho_j : j\in\mathcal{J}\}$. First, this collection of basis functions is availability-tracking. Second, ρ_j is non-decreasing in the sense that $\rho_j(\boldsymbol{x}) \ge \rho_j(\boldsymbol{x}-\boldsymbol{e}_i)$ for any $\boldsymbol{x}\in\mathcal{Q}$ with $x_i\ge 1$. Third, we show that $\Delta_{\mathscr{D}} \le \Delta_{\mathscr{B}}$. In the next lemma, we show that the collection of basis functions $\mathscr{D} = \{\rho_j : j\in\mathcal{J}\}$ is availability-tracking.

Lemma A.3 *If the collection of basis functions $\mathscr{B} = \{\varphi_j : j\in\mathcal{J}\}$ is availability-tracking, then the collection of basis functions $\mathscr{D} = \{\rho_j : j\in\mathcal{J}\}$ with ρ_j as in* (17) *is also availability-tracking.*

Proof: Since the collection $\mathscr{B} = \{\varphi_j : j\in\mathcal{J}\}$ is availability-tracking, it satisfies the three parts of Definition 3.1. We proceed to verify that the collection $\{\rho_j : j\in\mathcal{J}\}$ satisfies the three parts of

Definition 3.1 as well. To verify part (a), let $\boldsymbol{x} \in \mathcal{Q}$ be such that $x_i = 0$ for some $i \in A^j$. Since $x_i = 0$, we can write (17) as

$$\rho_j(\boldsymbol{x}) \quad = \max_{(\delta_\ell\,:\,\ell \in A^j \setminus \{i\}) \in \mathbb{Z}_+^{|A^j|-1}} \left\{ \varphi_j\left(\boldsymbol{x} - \sum_{\ell \in A^j \setminus \{i\}} \delta_\ell \, \boldsymbol{e}_\ell \right) \;:\; 0 \le \delta_\ell \le x_\ell \quad \forall\, \ell \in A^j \setminus \{i\} \right\}.$$

Because $x_i = 0$, the i-th component of the vector $\boldsymbol{x} - \sum_{\ell \in A^j \setminus \{i\}} \delta_\ell \, \boldsymbol{e}_\ell$ is zero, in which case, since φ_j satisfies part (a) and $i \in A^j$, we get $\varphi_j(\boldsymbol{x} - \sum_{\ell \in A \setminus \{i\}} \delta_\ell \, \boldsymbol{e}_\ell) = 0$ for any $(\delta_\ell : \ell \in A^j \setminus \{i\})$ that satisfies $0 \le \delta_\ell \le x_\ell$ for all $\ell \in A^j \setminus \{i\}$. Therefore, the objective value of the maximization problem on the right side above is zero at any feasible solution. Thus, $\rho_j(\boldsymbol{x}) = 0$, verifying part (a).

To verify part (b), let $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{Q}$ be such that $x_i = y_i$ for all $i \in A^j$. In this case, for each $i \in A^j$, the i-th components of the vectors $\boldsymbol{x} - \sum_{\ell \in A^j} \delta_\ell \, \boldsymbol{e}_\ell$ and $\boldsymbol{y} - \sum_{\ell \in A^j} \delta_\ell \, \boldsymbol{e}_\ell$ are the same. Therefore, because φ_j satisfies part (b), we get $\varphi_j(\boldsymbol{x} - \sum_{\ell \in A^j} \delta_\ell \, \boldsymbol{e}_\ell) = \varphi_j(\boldsymbol{y} - \sum_{\ell \in A^j} \delta_\ell \, \boldsymbol{e}_\ell)$. Once again, using the fact that $x_i = y_i$ for all $i \in A^j$, we get

$$\begin{aligned}
\rho_j(\boldsymbol{x}) \quad &= \max_{(\delta_\ell\,:\,\ell \in A^j) \in \mathbb{Z}_+^{|A^j|}} \left\{ \varphi_j\left(\boldsymbol{x} - \sum_{\ell \in A^j} \delta_\ell \, \boldsymbol{e}_\ell \right) \;:\; 0 \le \delta_\ell \le x_\ell \quad \forall\, \ell \in A^j \right\} \\
&= \max_{(\delta_\ell\,:\,\ell \in A^j) \in \mathbb{Z}_+^{|A^j|}} \left\{ \varphi_j\left(\boldsymbol{y} - \sum_{\ell \in A^j} \delta_\ell \, \boldsymbol{e}_\ell \right) \;:\; 0 \le \delta_\ell \le y_\ell \quad \forall\, \ell \in A^j \right\} \quad = \quad \rho_j(\boldsymbol{y}),
\end{aligned}$$

verifying part (b). To verify part (c), because $\varphi_j(\boldsymbol{x}) \le 1$ for any $\boldsymbol{x} \in \mathcal{Q}$, it holds that $\rho_j(\boldsymbol{C}) \le 1$ by (17). On the other hand, since setting $\delta_\ell = 0$ for all $\ell \in A^j$ is a feasible solution to the maximization problem in (17), $\rho_j(\boldsymbol{x}) \ge \varphi_j(\boldsymbol{x})$ for any $\boldsymbol{x} \in \mathcal{Q}$, in which case, since φ_j satisfies part (c), we get $\rho_j(\boldsymbol{C}) \ge \varphi_j(\boldsymbol{C}) = 1$. Thus, we have $\rho_j(\boldsymbol{C}) = 1$, verifying part (c). $\square$

Next, we show that ρ_j is non-decreasing.

Lemma A.4 *If φ_j is an availability-tracking basis function, then letting ρ_j be as in* (17)*, we have $\rho_j(\boldsymbol{x}) \ge \rho_j(\boldsymbol{x} - \boldsymbol{e}_i)$ for any $\boldsymbol{x} \in \mathcal{Q}$ with $x_i \ge 1$.*

Proof: First, assume that $i \in A^j$. Since $x_i \ge 1$, the interval $[1, x_i]$ is non-empty. Restricting the constraint $0 \le \delta_i \le x_i$ to $1 \le \delta_i \le x_i$ in problem (17), we get

$$\begin{aligned}
\rho_j(\boldsymbol{x}) \quad &= \max_{(\delta_\ell\,:\,\ell \in A^j) \in \mathbb{Z}_+^{|A^j|}} \left\{ \varphi_j\left(\boldsymbol{x} - \sum_{\ell \in A^j} \delta_\ell \, \boldsymbol{e}_\ell \right) \;:\; 0 \le \delta_\ell \le x_\ell \quad \forall\, \ell \in A^j \right\} \\
&\ge \max_{(\delta_\ell\,:\,\ell \in A^j) \in \mathbb{Z}_+^{|A^j|}} \left\{ \varphi_j\left(\boldsymbol{x} - \sum_{\ell \in A^j} \delta_\ell \, \boldsymbol{e}_\ell \right) \;:\; 0 \le \delta_\ell \le x_\ell \quad \forall\, \ell \in A^j \setminus \{i\},\ 1 \le \delta_i \le x_i \right\}.
\end{aligned}$$

We claim that the expression on the right side above is equal to $\rho_j(\boldsymbol{x} - \boldsymbol{e}_i)$, in which case, the result follows. To see the claim, we add and subtract $\boldsymbol{e}_i$ in the argument of φ_j on the right side above and do the change of variables $\gamma_i = \delta_i - 1$ and $\gamma_\ell = \delta_\ell$ for all $\ell \in A \setminus \{i\}$. Therefore, we can write the maximization problem on the right side above equivalently as

$$\begin{aligned}
&\max_{(\delta_\ell\,:\,\ell\in A^j)\in\mathbb{Z}_+^{|A^j|}} \left\{ \varphi_j\left(\boldsymbol{x} - \sum_{\ell\in A^j} \delta_\ell\, \boldsymbol{e}_\ell\right) \;:\; 0 \le \delta_\ell \le x_\ell \;\; \forall\, \ell \in A^j \setminus \{i\},\; 1 \le \delta_i \le x_i \right\} \\
&= \max_{(\delta_\ell\,:\,\ell\in A^j)\in\mathbb{Z}_+^{|A^j|}} \left\{ \varphi_j\left(\boldsymbol{x} - \boldsymbol{e}_i - \sum_{\ell\in A^j\setminus\{i\}} \delta_\ell\, \boldsymbol{e}_\ell - (\delta_i - 1)\,\boldsymbol{e}_i\right) \;:\; 0 \le \delta_\ell \le x_\ell \;\; \forall\, \ell \in A^j \setminus \{i\},\; 1 \le \delta_i \le x_i \right\} \\
&= \max_{(\gamma_\ell\,:\,\ell\in A^j)\in\mathbb{Z}_+^{|A^j|}} \left\{ \varphi_j\left(\boldsymbol{x} - \boldsymbol{e}_i - \sum_{\ell\in A^j\setminus\{i\}} \gamma_\ell\, \boldsymbol{e}_\ell - \gamma_i\,\boldsymbol{e}_i\right) \;:\; 0 \le \gamma_\ell \le x_\ell \;\; \forall\, \ell \in A^j \setminus \{i\},\; 0 \le \gamma_i \le x_i - 1 \right\} \\
&= \max_{(\gamma_\ell\,:\,\ell\in A^j)\in\mathbb{Z}_+^{|A^j|}} \left\{ \varphi_j\left(\boldsymbol{x} - \boldsymbol{e}_i - \sum_{\ell\in A^j} \gamma_\ell\, \boldsymbol{e}_\ell\right) \;:\; 0 \le \gamma_\ell \le x_\ell \;\; \forall\, \ell \in A^j \setminus \{i\},\; 0 \le \gamma_i \le x_i - 1 \right\} = \rho_j(\boldsymbol{x} - \boldsymbol{e}_i),
\end{aligned}$$

where the last equality is by (17). Thus, the claim holds. Second, assume that $i \notin A^j$. Since ρ_j is availability-tracking by Lemma A.3, by part (b) of Definition 3.1, we get $\rho_j(\boldsymbol{x}) = \rho_j(\boldsymbol{x} - \boldsymbol{e}_i)$. $\square$

Finally, we show that $\Delta_{\mathscr{D}} \le \Delta_{\mathscr{B}}$, where the collection $\mathscr{D} = \{\rho_j : j \in \mathcal{J}\}$ is constructed by using the collection $\mathscr{B} = \{\varphi_j : j \in \mathcal{J}\}$ as in (17).

Lemma A.5 *If the collection of basis functions $\mathscr{B} = \{\varphi_j : j \in \mathcal{J}\}$ is availability-tracking, then the collection of basis functions $\mathscr{D} = \{\rho_j : j \in \mathcal{J}\}$ with ρ_j as in (17) satisfies $\Delta_{\mathscr{D}} \le \Delta_{\mathscr{B}}$.*

Proof: To show the result in the lemma, for any $j \in \mathcal{J}$, $i \in \mathcal{L}$ and $\boldsymbol{x} \in \mathcal{Q}$ satisfying $x_i \ge 1$, we will establish the claim that

$$\rho_j(\boldsymbol{x}) - \rho_j(\boldsymbol{x} - \boldsymbol{e}_i) \le \max_{\boldsymbol{y}\in\mathcal{Q}:y_i\ge 1} \Big\{ \varphi_j(\boldsymbol{y}) - \varphi_j(\boldsymbol{y} - \boldsymbol{e}_i) \Big\}. \tag{18}$$

In this case, we get $\Delta_{\mathscr{D}} = \max_{j\in\mathcal{J}, i\in\mathcal{L}} C_i \times \max_{\boldsymbol{x}\in\mathcal{Q}:x_i\ge 1} \{\rho_j(\boldsymbol{x}) - \rho_j(\boldsymbol{x} - \boldsymbol{e}_i)\} \le \max_{j\in\mathcal{J}, i\in\mathcal{L}} C_i \times \max_{\boldsymbol{y}\in\mathcal{Q}:y_i\ge 1} \{\varphi_j(\boldsymbol{y}) - \varphi_j(\boldsymbol{y} - \boldsymbol{e}_i)\} = \Delta_{\mathscr{B}}$, which is the desired result. If $i \notin A^j$, then by part (b) of Definition 3.1, $\varphi_j(\boldsymbol{y}) = \varphi_j(\boldsymbol{y} - \boldsymbol{e}_i)$. Similarly, because ρ_j is availability-tracking by Lemma A.3, we have $\rho_j(\boldsymbol{x}) = \rho_j(\boldsymbol{x} - \boldsymbol{e}_i)$ as well. Therefore, both sides of the inequality in (18) are zero and the claim follows. In the rest of the proof, we assume that $i \in A^j$. By (17), we have

$$\begin{aligned}
\rho_j(\boldsymbol{x}) - \rho_j(\boldsymbol{x} - \boldsymbol{e}_i) \quad = &\max_{(\delta_\ell\,:\,\ell\in A^j)\in\mathbb{Z}_+^{|A^j|}} \left\{ \varphi_j\left(\boldsymbol{x} - \sum_{\ell\in A^j} \delta_\ell\, \boldsymbol{e}_\ell\right) \;:\; 0 \le \delta_\ell \le x_\ell \;\; \forall\, \ell \in A^j \right\} \\
&- \max_{(\delta_\ell\,:\,\ell\in A^j)\in\mathbb{Z}_+^{|A^j|}} \left\{ \varphi_j\left(\boldsymbol{x} - \boldsymbol{e}_i - \sum_{\ell\in A^j} \delta_\ell\, \boldsymbol{e}_\ell\right) \;:\; 0 \le \delta_\ell \le x_\ell \;\; \forall\, \ell \in A^j \setminus \{i\},\; 0 \le \delta_i \le x_i - 1 \right\}.
\end{aligned}$$

Let $\delta^* = (\delta^*_\ell : \ell \in A^j)$ be an optimal solution to the first maximization problem above. First, assume that $\delta^*_i < x_i$. In this case, $\delta^* = (\delta^*_\ell : \ell \in A^j)$ is a feasible but not necessarily optimal solution to the second maximization problem above. If we evaluate the objective function of the second maximization problem above at this feasible solution, then it follows that $\rho_j(\boldsymbol{x}) - \rho_j(\boldsymbol{x}-\boldsymbol{e}_i) \le \varphi_j\left(\boldsymbol{x} - \sum_{\ell\in A^j}\delta^*_\ell\,\boldsymbol{e}_\ell\right) - \varphi_j\left(\boldsymbol{x}-\boldsymbol{e}_i - \sum_{\ell\in A^j}\delta^*_\ell\,\boldsymbol{e}_\ell\right)$. Since $0 \le \delta^*_\ell \le x_\ell$ for all $\ell \in A^j$, we have $\boldsymbol{x} - \sum_{\ell\in A^j}\delta^*_\ell\,\boldsymbol{e}_\ell \in \mathcal{Q}$. In addition, since $\delta^*_i < x_i$, the i-th component of the vector $\boldsymbol{x} - \sum_{\ell\in A^j}\delta^*_\ell\,\boldsymbol{e}_\ell$ is at least one. Thus, the claim in (18) follows by

$$\rho_j(\boldsymbol{x}) - \rho_j(\boldsymbol{x}-\boldsymbol{e}_i) \le \varphi_j\left(\boldsymbol{x} - \sum_{\ell\in A^j}\delta^*_\ell\,\boldsymbol{e}_\ell\right) - \varphi_j\left(\boldsymbol{x}-\boldsymbol{e}_i - \sum_{\ell\in A^j}\delta^*_\ell\,\boldsymbol{e}_\ell\right) \le \max_{\boldsymbol{y}\in\mathcal{Q}:y_i\ge 1}\Big\{\varphi_j(\boldsymbol{y}) - \varphi_j(\boldsymbol{y}-\boldsymbol{e}_i)\Big\}.$$

Second, assume that $\delta^*_i = x_i$. So, the i-th component of the vector $\boldsymbol{x} - \sum_{\ell\in A^j}\delta^*_\ell\,\boldsymbol{e}_\ell$ is zero. Thus, by part (a) of Definition 3.1, we have $\varphi_j(\boldsymbol{x} - \sum_{\ell\in A^j}\delta^*_\ell\,\boldsymbol{e}_\ell) = 0$, so $\rho_j(\boldsymbol{x}) = \varphi_j(\boldsymbol{x} - \sum_{\ell\in A^j}\delta^*_\ell\,\boldsymbol{e}_\ell) = 0$, yielding $\rho_j(\boldsymbol{x}) - \rho_j(\boldsymbol{x}-\boldsymbol{e}_i) \le 0$. Also, $\varphi_j(\boldsymbol{C}) = 1 \ge \varphi_j(\boldsymbol{C}-\boldsymbol{e}_i)$, so $\max_{\boldsymbol{y}\in\mathcal{Q}:y_i\ge1}\{\varphi_j(\boldsymbol{y}) - \varphi_j(\boldsymbol{y}-\boldsymbol{e}_i)\} \ge 0$. Thus, we get $\rho_j(\boldsymbol{x}) - \rho_j(\boldsymbol{x}-\boldsymbol{e}_i) \le 0 \le \max_{\boldsymbol{y}\in\mathcal{Q}:y_i\ge1}\{\varphi_j(\boldsymbol{y}) - \varphi_j(\boldsymbol{y}-\boldsymbol{e}_i)\}$, establishing (18). $\square$

Letting $\varphi_j(\boldsymbol{x}) = \prod_{i\in A^j} f(\frac{x_i}{C_i})$ with $f(a) = \frac{2}{3} - \frac{5}{3}\,|a - \frac{2}{5}|$ for $a \le \frac{3}{5}$ and $f(a) = \frac{5}{3}\,(x - \frac{2}{5})$ for $a \ge \frac{3}{5}$, one can check that $\mathscr{B} = \{\varphi_j : j \in \mathcal{J}\}$ is availability-tracking with $\Delta_{\mathscr{B}} = \frac{5}{3}$ and φ_j is non-monotone.

Appendix B: Performance Guarantee under Customer Choice Behavior

In this section, we give a proof for Theorem 4.2 by using a sequence of lemmas. The next lemma shows that the coefficients $\{\gamma^t_j : j\in\mathcal{J},\ t\in\mathcal{T}\}$ computed through (12) are non-decreasing in t.

Lemma B.1 *For all $j\in\mathcal{J}$ and $t\in\mathcal{T}$, we have $\gamma^t_j \ge \gamma^{t+1}_j$.*

Proof: Considering problem (11), we define $\mu_j = r_j - \theta\sum_{i\in A^j}\frac{1}{C_i}\sum_{k\in\mathcal{J}}\mathbb{1}_{\{i\in A^k\}}\gamma^{t+1}_k$ for notational brevity. We claim that if $\hat{S}^t$ is an optimal solution to problem (11), then we have $\phi^t_j(\hat{S}^t)\,\mu_j \ge 0$ for all $j \in \hat{S}^t$. We prove this claim by contradiction. Suppose, on the contrary, that the set $\hat{N}^t = \{j\in\hat{S}^t : \phi^t_j(\hat{S}^t)\,\mu_j < 0\}$ is nonempty. Letting $\hat{P}^t = \{j\in\hat{S}^t\setminus\hat{N}^t : \mu_j \ge 0\}$, we have $\hat{P}^t \subseteq \hat{S}^t$, in which case, by Assumption 4.1, we have $\phi^t_j(\hat{P}^t) \ge \phi^t_j(\hat{S}^t)$ for all $j \in \hat{P}^t$. Moreover, by the same assumption, since $\hat{S}^t\in\mathcal{F}$, we have $\hat{P}^t\in\mathcal{F}$ as well. So, we obtain the chain of inequalities

$$\sum_{j\in\hat{S}^t}\phi^t_j(\hat{S}^t)\,\mu_j = \sum_{j\in\hat{S}^t\setminus\hat{N}^t}\phi^t_j(\hat{S}^t)\,\mu_j + \sum_{j\in\hat{N}^t}\phi^t_j(\hat{S})\,\mu_j < \sum_{j\in\hat{S}^t\setminus\hat{N}^t}\phi^t_j(\hat{S}^t)\,\mu_j \le \sum_{j\in\hat{P}^t}\phi^t_j(\hat{S}^t)\,\mu_j \le \sum_{j\in\hat{P}^t}\phi^t_j(\hat{P}^t)\,\mu_j,$$

where the first inequality uses the fact that $\phi^t_j(\hat{S}^t)\,\mu_j < 0$ for all $j\in\hat{N}^t$ and $\hat{N}^t\neq\varnothing$, the second inequality uses the fact that $\mu_j < 0$ whenever $j\in\hat{S}^t\setminus\hat{N}^t$ and $j\notin\hat{P}^t$ and the third inequality uses the fact that $\phi^t_j(\hat{P}^t)\ge\phi^t_j(\hat{S}^t)$ and $\mu_j\ge0$ for all $j\in\hat{P}^t$. Since $\hat{P}^t\in\mathcal{F}$, the solution $\hat{P}^t$ is feasible to problem (11), so the chain of inequalities above contradicts the fact that $\hat{S}^t$ is an optimal solution to problem (11), establishing the claim. In this case, if $j\in\hat{S}^t$, then (12) implies that $\gamma^t_j = \gamma^{t+1}_j + \phi^t_j(\hat{S}^t)\,\mu_j \ge \gamma^{t+1}_j$. If $j\notin\hat{S}^t$, then we have $\phi^t_j(\hat{S}^t) = 0$, so $\gamma^t_j = \gamma^{t+1}_j + \phi^t_j(\hat{S}^t)\,\mu_j = \gamma^{t+1}_j$. $\square$

The next lemma gives a bound on the optimal total expected revenue obtained through (10).

Lemma B.2 *If the coefficients* $\{\gamma_j^t : j \in \mathcal{J},\ t \in \mathcal{T}\}$ *are computed through* (12)*, then we have* $V^1(\boldsymbol{C}) \le (1+\theta L)\sum_{j\in\mathcal{J}} \gamma_j^1$.

Proof: We formulate a linear program whose optimal objective value provides an upper bound on the optimal total expected revenue. In particular, we use the vector of decision variables $\boldsymbol{z} = (z^t(S) : S \in \mathcal{F},\ t \in \mathcal{T})$, where $z^t(S)$ is the probability that we offer the assortment S at time period t. In this case, we consider the linear program

$$\mathsf{Z}^*_{LP} = \max_{\boldsymbol{z}\in\mathbb{R}_+^{|\mathcal{F}|\times T}} \left\{ \sum_{t\in\mathcal{T}}\sum_{S\in\mathcal{F}}\sum_{j\in\mathcal{J}} r_j\, \phi_j^t(S)\, z^t(S) \ : \ \sum_{t\in\mathcal{T}}\sum_{S\in\mathcal{F}}\sum_{j\in\mathcal{J}} \mathbb{1}_{\{i\in A^j\}}\, \phi_j^t(S)\, z^t(S) \le C_i \quad \forall\, i \in \mathcal{L}, \right.$$
$$\left. \sum_{S\in\mathcal{F}} z^t(S) = 1 \quad \forall\, t \in \mathcal{T} \right\}. \tag{19}$$

The objective function above accounts for the total expected revenue over the selling horizon. The first constraint ensures that the total expected capacity consumption of the resources do not exceed the initial capacities. The second constraint ensures that we offer an assortment at each time period with probability one. It is well-known that the optimal objective value of problem (19) is an upper bound on the optimal total expected revenue; see Liu and van Ryzin (2008). Thus, we have $\mathsf{Z}^*_{LP} \ge V^1(\boldsymbol{C})$. By Assumption 4.1, it is simple to see that $\varnothing \in \mathcal{F}$. Offering the empty assortment at each time period yields a feasible solution to problem (19). By the second constraint above, the decision variables are upper bounded by one. Therefore, problem (19) is feasible and bounded, so the optimal objective value of the dual of problem (19) is also Z^*_{LP}. Using the vectors of dual variables $\boldsymbol{\mu} = (\mu_i : i \in \mathcal{L})$ and $\boldsymbol{\sigma} = (\sigma^t : t \in \mathcal{T})$, the dual of problem (19) involves minimizing the objective function $\sum_{i\in\mathcal{L}} C_i\,\mu_i + \sum_{t\in\mathcal{T}} \sigma^t$ subject to the constraint $\sum_{i\in\mathcal{L}}\sum_{j\in\mathcal{J}} \mathbb{1}_{\{i\in A^j\}}\,\phi_j^t(S)\,\mu_i + \sigma^t \ge \sum_{j\in\mathcal{J}} r_j\,\phi_j^t(S)$ for all $S \in \mathcal{F},\ t \in \mathcal{T}$ with the decision variables $(\boldsymbol{\mu},\boldsymbol{\sigma}) \in \mathbb{R}_+^{|\mathcal{L}|} \times \mathbb{R}^T$. Since the decision variable σ^t appears with a positive coefficient in the objective function, it takes on its smallest possible value in an optimal solution. If we express the constraint as $\sigma^t \ge \sum_{j\in\mathcal{J}} \phi_j^t(S)\,(r_j - \sum_{i\in\mathcal{L}} \mathbb{1}_{\{i\in A^j\}}\,\mu_i)$ for all $S \in \mathcal{F},\ t \in \mathcal{T}$, then the smallest possible value of σ^t is $\max_{S\in\mathcal{F}} \sum_{j\in\mathcal{J}} \phi_j^t(S)\,(r_j - \sum_{i\in\mathcal{L}} \mathbb{1}_{\{i\in A^j\}}\,\mu_i)$, in which case, replacing the decision variable σ^t by its value in an optimal solution, the dual of problem (19) takes the form

$$\mathsf{Z}^*_{LP} = \min_{\boldsymbol{\mu}\in\mathbb{R}_+^{|\mathcal{L}|}} \left\{ \sum_{i\in\mathcal{L}} C_i\,\mu_i + \sum_{t\in\mathcal{T}} \max_{S\in\mathcal{F}} \left\{ \sum_{j\in\mathcal{J}} \phi_j^t(S) \left(r_j - \sum_{i\in\mathcal{L}} \mathbb{1}_{\{i\in A^j\}}\,\mu_i \right) \right\} \right\}. \tag{20}$$

Using the coefficients $\{\gamma_j^t : j \in \mathcal{J},\ t \in \mathcal{T}\}$ that are computed through (12), we construct a solution $\hat{\boldsymbol{\mu}} = (\hat{\mu}_i : i \in \mathcal{L})$ to problem (20) by setting $\hat{\mu}_i = \frac{\theta}{C_i} \sum_{k\in\mathcal{J}} \mathbb{1}_{\{i\in A^k\}}\,\gamma_k^1$ for all $i \in \mathcal{L}$. By Lemma B.1, we have $\gamma_j^1 \ge \gamma_j^2 \ge \ldots \ge \gamma_j^{T+1} = 0$. Therefore, we have $\hat{\mu}_i \ge 0$ for all $i \in \mathcal{L}$, so the solution $\hat{\boldsymbol{\mu}}$ is feasible to problem (20). In this case, noting that problem (20) is a minimization problem, the objective value of this problem evaluated at the solution $\hat{\boldsymbol{\mu}}$ is an upper bound on the optimal objective value Z^*_{LP}. Evaluating the objective function of problem (20) at the solution $\hat{\boldsymbol{\mu}}$, we get

$$\begin{aligned}
\mathsf{Z}^*_{LP} &\leq \sum_{i\in\mathcal{L}} C_i\,\hat{\mu}_i + \sum_{t\in\mathcal{T}} \max_{S\in\mathcal{F}} \left\{ \sum_{j\in\mathcal{J}} \phi^t_j(S) \left(r_j - \sum_{i\in\mathcal{L}} \mathbb{1}_{\{i\in A^j\}}\,\hat{\mu}_i \right) \right\} \\
&= \sum_{i\in\mathcal{L}} C_i \frac{\theta}{C_i} \sum_{k\in\mathcal{J}} \mathbb{1}_{\{i\in A^k\}}\,\gamma^1_k + \sum_{t\in\mathcal{T}} \max_{S\in\mathcal{F}} \left\{ \sum_{j\in\mathcal{J}} \phi^t_j(S) \left(r_j - \sum_{i\in\mathcal{L}} \mathbb{1}_{\{i\in A^j\}}\,\frac{\theta}{C_i} \sum_{k\in\mathcal{J}} \mathbb{1}_{\{i\in A^k\}}\,\gamma^1_k \right) \right\} \\
&= \theta \sum_{k\in\mathcal{J}} \sum_{i\in\mathcal{L}} \mathbb{1}_{\{i\in A^k\}}\,\gamma^1_k + \sum_{t\in\mathcal{T}} \max_{S\in\mathcal{F}} \left\{ \sum_{j\in\mathcal{J}} \phi^t_j(S) \left(r_j - \theta \sum_{i\in A^j} \frac{1}{C_i} \sum_{k\in\mathcal{J}} \mathbb{1}_{\{i\in A^k\}}\,\gamma^1_k \right) \right\} \\
&\leq \theta L \sum_{k\in\mathcal{J}} \gamma^1_k + \sum_{t\in\mathcal{T}} \max_{S\in\mathcal{F}} \left\{ \sum_{j\in\mathcal{J}} \phi^t_j(S) \left(r_j - \theta \sum_{i\in A^j} \frac{1}{C_i} \sum_{k\in\mathcal{J}} \mathbb{1}_{\{i\in A^k\}}\,\gamma^{t+1}_k \right) \right\} \\
&= \theta L \sum_{k\in\mathcal{J}} \gamma^1_k + \sum_{t\in\mathcal{T}} \sum_{j\in\mathcal{J}} \phi^t_j(\hat{S}^t) \left(r_j - \theta \sum_{i\in A^j} \frac{1}{C_i} \sum_{k\in\mathcal{J}} \mathbb{1}_{\{i\in A^k\}}\,\gamma^{t+1}_k \right) \\
&= \theta L \sum_{k\in\mathcal{J}} \gamma^1_k + \sum_{t\in\mathcal{T}} \sum_{j\in\mathcal{J}} \left[\gamma^t_j - \gamma^{t+1}_j \right] \quad = \quad (1+\theta L) \sum_{j\in\mathcal{J}} \gamma^1_j,
\end{aligned}$$

where the second inequality uses the fact that $\sum_{i\in\mathcal{L}} \mathbb{1}_{\{i\in A^k\}} = |A^k| \leq L$ and $\gamma^1_k \geq \gamma^2_k \geq \ldots \geq \gamma^T_k \geq \gamma^{T+1}_k$ for all $k\in\mathcal{J}$, the third equality holds by the definition of $\hat{S}^t$ in (11), the fourth equality holds because $\gamma^t_j - \gamma^{t+1}_j = \phi^t_j(\hat{S}^t)\left(r_j - \theta \sum_{i\in A^j} \frac{1}{C_i} \sum_{k\in\mathcal{J}} \mathbb{1}_{\{i\in A^k\}} \gamma^{t+1}_k\right)$ by (12). Thus, $(1+\theta L)\sum_{j\in\mathcal{J}} \gamma^1_j \geq \mathsf{Z}^*_{LP}$, in which case, the desired result follows from the fact that $\mathsf{Z}^*_{LP} \geq V^1(\boldsymbol{C})$. $\square$

Next, we give a lower bound on the total expected revenue obtained by our approximate policy. Let $U^t(\boldsymbol{x})$ denote the total expected revenue obtained by our approximate policy over time periods $t, t+1, \ldots, T$, given that the state of the resources at time period t is $\boldsymbol{x}$. Building on the dynamic program in (10), we can compute $\{U^t : t\in\mathcal{T}\}$ through the recursion

$$\begin{aligned}
U^t(\boldsymbol{x}) &= \sum_{j\in\mathcal{J}} \phi^t_j(S^{\mathsf{App},t}(\boldsymbol{x})) \left(\prod_{i\in A^j} \mathbb{1}_{\{x_i\geq 1\}} \right) \left[r_j + U^{t+1}\left(\boldsymbol{x} - \sum_{i\in A^j} \boldsymbol{e}_i \right) \right] \\
&\qquad + \left(1 - \sum_{j\in\mathcal{J}} \phi^t_j(S^{\mathsf{App},t}(\boldsymbol{x})) + \sum_{j\in\mathcal{J}} \phi^t_j(S^{\mathsf{App},t}(\boldsymbol{x})) \left(1 - \prod_{i\in A^j} \mathbb{1}_{\{x_i\geq 1\}} \right) \right) U^{t+1}(\boldsymbol{x}) \\
&= U^{t+1}(\boldsymbol{x}) + \sum_{j\in\mathcal{J}} \phi^t_j(S^{\mathsf{App},t}(\boldsymbol{x})) \left(\prod_{i\in A^j} \mathbb{1}_{\{x_i\geq 1\}} \right) \left[r_j - U^{t+1}(\boldsymbol{x}) + U^{t+1}\left(\boldsymbol{x} - \sum_{i\in A^j} \boldsymbol{e}_i \right) \right], \qquad (21)
\end{aligned}$$

with the boundary condition that $U^{T+1} = 0$. Recall that if the state of the remaining resources at time period t is $\boldsymbol{x}$, then the approximate policy offers the assortment $S^{\mathsf{App},t}(\boldsymbol{x})$ given in (13). The recursion above is analogous to the dynamic program in (10). The only difference is that the offered assortment is fixed by the decision of the approximate policy.

The total expected revenue obtained by our approximate policy is given by $U^1(\boldsymbol{C})$. In the next lemma, we give a lower bound on this quantity.

Lemma B.3 *If the tuning parameter θ satisfies $\theta \geq \Delta_{\mathscr{B}}$ and the coefficients $\{\gamma_j^t : j \in \mathcal{J},\ t \in \mathcal{T}\}$ are computed through* (12)*, then we have $U^1(\boldsymbol{C}) \geq \sum_{j\in\mathcal{J}} \gamma_j^1$.*

Proof: We use induction over the time periods to show that $U^t(\boldsymbol{x}) \geq H^t(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{Q}$ and $t \in \mathcal{T}$, where $H^t(\boldsymbol{x}) = \sum_{j\in\mathcal{J}} \gamma_j^t \varphi_j(\boldsymbol{x})$ with $\{\gamma_j^t : j \in \mathcal{J},\ t \in \mathcal{T}\}$ computed through (12). Consider the base case at time period $T+1$. Since $U^{T+1} = 0 = H^{T+1}$, the base case holds. Suppose that the result holds at time period $t+1$, so that $U^{t+1}(\boldsymbol{x}) \geq H^{t+1}(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{Q}$. We proceed to showing that $U^t(\boldsymbol{x}) \geq H^t(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{Q}$. Since $U^{t+1} \geq H^{t+1}$, if we replace U^{t+1} on the right side of the second equality in (21) with H^{t+1}, then by precisely the same line of reasoning at the beginning of the proof of Lemma 3.8, the right side of (21) gets smaller. Thus, we have

$$\begin{aligned}
U^t(\boldsymbol{x}) \;&\geq\; H^{t+1}(\boldsymbol{x}) + \sum_{j\in\mathcal{J}} \phi_j^t(S^{\mathsf{App},t}(\boldsymbol{x})) \left(\prod_{i\in A^j} \mathbb{1}_{\{x_i \geq 1\}}\right) \left[r_j - H^{t+1}(\boldsymbol{x}) + H^{t+1}\left(\boldsymbol{x} - \sum_{i\in A^j} \boldsymbol{e}_i\right)\right] \\
&=\; H^{t+1}(\boldsymbol{x}) + \max_{S\in\mathcal{F}} \left\{ \sum_{j\in\mathcal{J}} \phi_j^t(S) \left(\prod_{i\in A^j} \mathbb{1}_{\{x_i \geq 1\}}\right) \left[r_j - H^{t+1}(\boldsymbol{x}) + H^{t+1}\left(\boldsymbol{x} - \sum_{i\in A^j} \boldsymbol{e}_i\right)\right]\right\} \\
&\geq\; H^{t+1}(\boldsymbol{x}) + \max_{S\in\mathcal{F}} \left\{ \sum_{j\in\mathcal{J}} \phi_j^t(S) \left(\prod_{i\in A^j} \mathbb{1}_{\{x_i \geq 1\}}\right) \left[r_j - \theta \sum_{i\in A^j} \frac{1}{C_i} \sum_{k\in\mathcal{J}} \mathbb{1}_{\{i\in A^k\}} \gamma_k^{t+1}\right]\right\} \\
&\geq\; H^{t+1}(\boldsymbol{x}) + \sum_{j\in\mathcal{J}} \phi_j(\hat{S}^t) \left(\prod_{i\in A^j} \mathbb{1}_{\{x_i \geq 1\}}\right) \left[r_j - \theta \sum_{i\in A^j} \frac{1}{C_i} \sum_{k\in\mathcal{J}} \mathbb{1}_{\{i\in A^k\}} \gamma_k^{t+1}\right] \\
&=\; H^{t+1}(\boldsymbol{x}) + \sum_{j\in\mathcal{J}} \left(\prod_{i\in A^j} \mathbb{1}_{\{x_i \geq 1\}}\right) \left(\gamma_j^t - \gamma_j^{t+1}\right) \\
&\geq\; H^{t+1}(\boldsymbol{x}) + \sum_{j\in\mathcal{J}} \varphi_j(\boldsymbol{x}) \left(\gamma_j^t - \gamma_j^{t+1}\right) \;=\; H^t(\boldsymbol{x}).
\end{aligned}$$

Here, the first equality follows from the decision function for the approximate policy given in (13). The second inequality is by Lemma 3.7. The third inequality uses the fact that $\hat{S}^t \in \mathcal{F}$ is a feasible but not necessarily optimal solution to the maximization problem on the left side of the third inequality. The second equality is by (12). The fourth inequality follows by noting part (a) of Definition 3.1 and using the fact that $\gamma_j^t \geq \gamma_j^{t+1}$ by Lemma B.1. The last equality holds since $H^t(\boldsymbol{x}) = \sum_{j\in\mathcal{J}} \gamma_j^t \varphi_j(\boldsymbol{x})$. Thus, the induction argument is complete, so $U^t(\boldsymbol{x}) \geq H^t(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{Q}$, $t \in \mathcal{T}$. Using this inequality with $t = 1$ and $\boldsymbol{x} = \boldsymbol{C}$, we get $U^1(\boldsymbol{C}) \geq H^1(\boldsymbol{C}) = \sum_{j\in\mathcal{J}} \gamma_j^1$. □

Next, we use the two lemmas above to give a proof for Theorem 4.2.

Proof of Theorem 4.2: By Lemma B.2, we have $(1+\theta L)\sum_{j\in\mathcal{J}} \gamma_j^1 \geq V^1(\boldsymbol{C})$, whereas by Lemma B.3, we have $U^1(\boldsymbol{C}) \geq \sum_{j\in\mathcal{J}} \gamma_j^1$. Therefore, we get $U^1(\boldsymbol{C}) \geq \sum_{j\in\mathcal{J}} \gamma_j^1 \geq V^1(\boldsymbol{C})/(1+\theta L)$. □

Appendix C: Performance Guarantee under Multiple Units of Capacity Consumption

In this section, we give a proof for Theorem 4.3. Recall that product j uses a_{ij} units of resource i. We still use $A^j = \{i \in \mathcal{L} : a_{ij} \geq 1\}$ to denote the set of resources used by product j and $L = \max_{j\in\mathcal{J}} |A^j|$ to denote the maximum number of resources used by a product. Lastly, the maximum number of units of resource i that is used by any product is given by $m_i = \max_{j\in\mathcal{J}} a_{ij}$. As discussed at the beginning of Section 4.2, we can compute the optimal policy by solving a dynamic program that is similar to the one in (1). All we need to do is to replace all occurrences of $\prod_{i\in A^j} \mathbb{1}_{\{x_i\geq 1\}}$ with $\prod_{i\in A^j} \mathbb{1}_{\{x_i\geq a_{ij}\}}$ and all occurrences of $\sum_{i\in A^j} \boldsymbol{e}_i$ with $\sum_{i\in A^j} a_{ij}\, \boldsymbol{e}_i$ in the dynamic program. In particular, we can compute the optimal policy by computing the value functions $\{V^t : t \in \mathcal{T}\}$ through the dynamic program

$$V^t(\boldsymbol{x}) \;=\; V^{t+1}(\boldsymbol{x}) + \sum_{j\in\mathcal{J}} \lambda_j^t \left(\prod_{i\in A^j} \mathbb{1}_{\{x_i\geq a_{ij}\}}\right) \left[r_j - V^{t+1}(\boldsymbol{x}) + V^{t+1}\left(\boldsymbol{x} - \sum_{i\in A^j} a_{ij}\boldsymbol{e}_i\right)\right]^+,$$

with the boundary condition that $V^{T+1} = 0$. If $r_j \geq V^{t+1}(\boldsymbol{x}) - V^{t+1}(\boldsymbol{x} - \sum_{i\in A^j} a_{ij}\, \boldsymbol{e}_i)$, then it is optimal to accept a request for product j, as long as we have $\prod_{i\in A^j} \mathbb{1}_{\{x_i\geq a_{ij}\}} = 1$.

The next lemma generalizes Lemma 3.7 to the case where each product can use multiple units of a resource. Recall that the mapping $\mathtt{G}_j : \mathcal{Q} \to \mathcal{Q}$ is given by $\mathtt{G}_j(\boldsymbol{x}) = \big(x_i\, \mathbb{1}_{\{x_i\geq a_{ij}\}} : i \in \mathcal{L}\big)$.

Lemma C.1 *For a collection of availability-tracking basis functions $\mathscr{B} = \{\varphi_j : j \in \mathcal{J}\}$, let $H^t(\boldsymbol{x}) = \sum_{k\in\mathcal{J}} \gamma_k^t\, \varphi_k(\mathtt{G}_k(\boldsymbol{x}))$, where the coefficients $\{\gamma_k^t : k \in \mathcal{J}\}$ satisfy $\gamma_k^t \geq 0$ for all $k \in \mathcal{J}$. Then, for each $j \in \mathcal{J}$ and $\boldsymbol{x} \in \mathcal{Q}$ such that $\boldsymbol{x} - \sum_{i\in A^j} a_{ij}\, \boldsymbol{e}_i \geq \boldsymbol{0}$, we have*

$$H^t(\boldsymbol{x}) - H^t\left(\boldsymbol{x} - \sum_{i\in A^j} a_{ij}\, \boldsymbol{e}_i\right) \;\leq\; \Delta_{\mathscr{B}} \sum_{i\in A^j} \frac{2m_i - 1}{C_i} \sum_{k\in\mathcal{J}} \mathbb{1}_{\{i\in A^k\}}\, \gamma_k^t.$$

Proof: Let $\boldsymbol{y} = \mathtt{G}_k(\boldsymbol{x})$ and $\tilde{\boldsymbol{y}} = \mathtt{G}_k(\boldsymbol{x} - a_{ij}\, \boldsymbol{e}_i)$. By the definition of $\mathtt{G}_k$, note that $\boldsymbol{y}$ and $\tilde{\boldsymbol{y}}$ can differ only in the i-th component. Moreover, by the definition of $\mathtt{G}_k$, we have

$$y_i - \tilde{y}_i = x_i\, \mathbb{1}_{\{x_i\geq a_{ik}\}} - (x_i - a_{ij})\, \mathbb{1}_{\{x_i - a_{ij}\geq a_{ik}\}} = \begin{cases} a_{ij} & \text{if } \; x_i \geq a_{ij} + a_{ik} \\ x_i & \text{if } \; a_{ij} + a_{ik} - 1 \geq x_i \geq a_{ik} \\ 0 & \text{if } \; a_{ik} - 1 \geq x_i, \end{cases}$$

which implies that y_i and $\tilde{y}_i$ differ from each other by at most $a_{ij} + a_{ik} - 1$, as long as $i \in A^k$ so that $a_{ik} \geq 1$. By the definition of m_i, $a_{ij} + a_{ik} - 1 \leq 2\, m_i - 1$. So, by a telescoping sum, we get

$$\varphi_k(\boldsymbol{y}) - \varphi_k(\tilde{\boldsymbol{y}}) = \sum_{h=1}^{y_i - \tilde{y}_i} \left(\varphi_k(\tilde{\boldsymbol{y}} + h\, \boldsymbol{e}_i) - \varphi_k(\tilde{\boldsymbol{y}} + (h-1)\, \boldsymbol{e}_i)\right) \;\leq \frac{\Delta_{\mathscr{B}}(2m_i - 1)}{C_i}, \tag{22}$$

where the inequality follows from the definition of $\Delta_{\mathscr{B}}$. We show the inequality in the lemma by using induction on the cardinality of A^j. Consider the base case where $|A^j| = 1$ so that $A^j = \{i\}$

for some $i \in \mathcal{L}$. Since $\mathtt{G}_k(\boldsymbol{x})$ and $\mathtt{G}_k(\boldsymbol{x} - a_{ij}\,\boldsymbol{e}_i)$ differ only in the i-th component, by part (b) of Definition 3.1, we have $\varphi_k(\mathtt{G}_k(\boldsymbol{x})) - \varphi_k(\mathtt{G}_k(\boldsymbol{x} - a_{ij}\boldsymbol{e}_i)) = 0$ whenever $i \notin A^k$. So, we get

$$\begin{aligned} H^t(\boldsymbol{x}) - H^t\left(\boldsymbol{x} - a_{ij}\boldsymbol{e}_i\right) &= \sum_{k\in\mathcal{J}} \mathbb{1}_{\{i\in A^k\}}\gamma_k^t\Big(\varphi_k(\mathtt{G}_k(\boldsymbol{x})) - \varphi_k(\mathtt{G}_k(\boldsymbol{x} - a_{ij}\boldsymbol{e}_i))\Big) \\ &\le \frac{\Delta_{\mathscr{B}}(2m_i - 1)}{C_i}\sum_{k\in\mathcal{J}} \mathbb{1}_{\{i\in A^k\}}\gamma_k^t, \end{aligned}$$

where the inequality is by (22). Thus, the base case holds. Suppose that the results holds for any $|A^j| \le s$. Consider the case where $|A^j| = s+1$ so that $A^j = B \cup \{\ell\}$ for some $B \subseteq \mathcal{L}$ with $|B| = s$ and $\ell \in \mathcal{L}$ with $\ell \notin B$. Letting $\boldsymbol{w} = \boldsymbol{x} - \sum_{i\in B} a_{ij}\,\boldsymbol{e}_i$ for notational brevity, we have

$$\begin{aligned} H^t(\boldsymbol{x}) - H^t\left(\boldsymbol{x} - \sum_{i\in A^j} a_{ij}\,\boldsymbol{e}_i\right) &= H^t(\boldsymbol{x}) - H^t\left(\boldsymbol{x} - \sum_{i\in B} a_{ij}\,\boldsymbol{e}_i - a_{\ell j}\,\boldsymbol{e}_\ell\right) \\ &= H^t(\boldsymbol{x}) - H^t\left(\boldsymbol{x} - \sum_{i\in B} a_{ij}\,\boldsymbol{e}_i\right) + H^t\left(\boldsymbol{x} - \sum_{i\in B} a_{ij}\,\boldsymbol{e}_i\right) - H^t\left(\boldsymbol{x} - \sum_{i\in B} a_{ij}\,\boldsymbol{e}_i - a_{\ell j}\,\boldsymbol{e}_\ell\right) \\ &= H^t(\boldsymbol{x}) - H^t\left(\boldsymbol{x} - \sum_{i\in B} a_{ij}\,\boldsymbol{e}_i\right) + H^t(\boldsymbol{w}) - H^t(\boldsymbol{w} - a_{\ell j}\,\boldsymbol{e}_\ell) \\ &\le \Delta_{\mathscr{B}}\sum_{i\in B}\frac{2m_i - 1}{C_i}\sum_{k\in\mathcal{J}} \mathbb{1}_{\{i\in A^k\}}\gamma_k^t \;+\; H^t\left(\boldsymbol{w}\right) - H^t\left(\boldsymbol{w} - a_{\ell j}\,\boldsymbol{e}_\ell\right) \\ &\le \Delta_{\mathscr{B}}\sum_{i\in B}\frac{2m_i - 1}{C_i}\sum_{k\in\mathcal{J}} \mathbb{1}_{\{i\in A^k\}}\gamma_k^t \;+\; \frac{\Delta_{\mathscr{B}}(2m_\ell - 1)}{C_\ell}\sum_{k\in\mathcal{J}} \mathbb{1}_{\{\ell\in A^k\}}\gamma_k^t \\ &= \Delta_{\mathscr{B}}\sum_{i\in A^j}\frac{2m_i - 1}{C_i}\sum_{k\in\mathcal{J}} \mathbb{1}_{\{i\in A^k\}}\gamma_k^t, \end{aligned}$$

where the first inequality is by the induction assumption along the fact that $|B| = s$, whereas the second inequality is by the base case. Thus, the induction argument is complete. □

The next lemma gives an upper bound on the optimal total expected revenue. It generalizes Lemma 3.6 to multiple units of resource consumption by the products.

Lemma C.2 *If the coefficients* $\{\gamma_j^t : j \in \mathcal{J},\ t \in \mathcal{T}\}$ *are computed through* (15)*, then we have* $V^1(\boldsymbol{C}) \le (1 + \theta\,(2M-1)L)\sum_{j\in\mathcal{J}}\gamma_j^1$.

Proof: Considering the linear program in (4), if we replace all occurrences of $\mathbb{1}_{\{i\in A^j\}}$ with a_{ij}, then the optimal objective value of this linear program provides an upper bound on the optimal total expected revenue when a product can consume multiple units of a resource. By following precisely the same sequence of steps that we use in the proof of Lemma 3.6, we can obtain the optimal objective value of this linear program by solving its dual and the dual of this linear program can be obtained by replacing all occurrences of $\mathbb{1}_{\{i\in A^j\}}$ in the linear program in (6) with a_{ij}. Therefore, we can obtain an upper bound on the optimal total expected revenue by using the optimal objective value of the linear program

$$\mathsf{Z}^*_{LP} \;=\; \min_{\boldsymbol{\mu}\in\mathbb{R}_+^{|\mathcal{L}|}} \left\{ \sum_{i\in\mathcal{L}} C_i\,\mu_i + \sum_{j\in\mathcal{J}} \Lambda_j \left[r_j - \sum_{i\in\mathcal{L}} a_{ij}\,\mu_i \right]^+ \right\}, \tag{23}$$

where we recall that $\Lambda_j = \sum_{t\in\mathcal{T}} \lambda_j^t$. We construct a solution $\hat{\boldsymbol{\mu}} = (\hat{\mu} : i\in\mathcal{L})$ to the problem above by setting $\hat{\mu}_i = \frac{\theta\,(2m_i-1)}{C_i} \sum_{k\in\mathcal{J}} \mathbb{1}_{\{i\in A^k\}}\,\gamma_k^1$ for all $i\in\mathcal{L}$. Since $[a]^+\geq 0$ for all $a\in\mathbb{R}$, by (15), we have $\gamma_j^1 \geq \gamma_j^2 \geq \ldots \geq \gamma_j^T \geq \gamma_j^{T+1} = 0$. Thus, $\hat{\mu}_i \geq 0$ for all $i\in\mathcal{L}$, which implies that $\hat{\boldsymbol{\mu}}$ is a feasible solution to problem (23). In this case, the objective value provided by the solution $\hat{\boldsymbol{\mu}}$ for the problem above is an upper bound on the optimal objective value Z^*_{LP}. Thus, evaluating the objective function of problem (23) at the solution $\hat{\boldsymbol{\mu}}$, we get

$$\begin{aligned}
\mathsf{Z}^*_{LP} \;&\leq\; \sum_{i\in\mathcal{L}} C_i\,\hat{\mu}_i + \sum_{j\in\mathcal{J}} \Lambda_j \left[r_j - \sum_{i\in\mathcal{L}} a_{ij}\,\hat{\mu}_i \right]^+ \\
&=\; \sum_{i\in\mathcal{L}} C_i \frac{\theta\,(2m_i-1)}{C_i} \sum_{k\in\mathcal{J}} \mathbb{1}_{\{i\in A^k\}}\,\gamma_k^1 + \sum_{j\in\mathcal{J}} \Lambda_j \left[r_j - \sum_{i\in\mathcal{L}} a_{ij}\,\frac{\theta\,(2m_i-1)}{C_i} \sum_{k\in\mathcal{J}} \mathbb{1}_{\{i\in A^k\}}\,\gamma_k^1 \right]^+ \\
&\leq\; \theta \sum_{k\in\mathcal{J}} \gamma_k^1 \sum_{i\in\mathcal{L}} \mathbb{1}_{\{i\in A^k\}}\,(2m_i-1) + \sum_{j\in\mathcal{J}} \Lambda_j \left[r_j - \sum_{i\in A^j} \frac{\theta\,(2m_i-1)}{C_i} \sum_{k\in\mathcal{J}} \mathbb{1}_{\{i\in A^k\}}\,\gamma_k^1 \right]^+ \\
&\leq\; \theta \sum_{k\in\mathcal{J}} \gamma_k^1 \sum_{i\in\mathcal{L}} \mathbb{1}_{\{i\in A^k\}}\,(2m_i-1) + \sum_{j\in\mathcal{J}} \sum_{t\in\mathcal{T}} \lambda_j^t \left[r_j - \sum_{i\in A^j} \frac{\theta\,(2m_i-1)}{C_i} \sum_{k\in\mathcal{J}} \mathbb{1}_{\{i\in A^k\}}\,\gamma_k^{t+1} \right]^+ \\
&=\; \theta \sum_{k\in\mathcal{J}} \gamma_k^1 \sum_{i\in\mathcal{L}} \mathbb{1}_{\{i\in A^k\}}\,(2m_i-1) + \sum_{j\in\mathcal{J}} \sum_{t\in\mathcal{T}} \left[\gamma_j^t - \gamma_j^{t+1} \right] \\
&\leq\; \theta\,(2M-1)\,L \sum_{k\in\mathcal{J}} \gamma_k^1 + \sum_{j\in\mathcal{J}} \sum_{t\in\mathcal{T}} \left[\gamma_j^t - \gamma_j^{t+1} \right] \\
&=\; (1+\theta\,(2M-1)L) \sum_{j\in\mathcal{J}} \gamma_j^1.
\end{aligned}$$

In the chain of inequalities above, the second inequality uses the fact that $a_{ij}\geq 1$ for all $i\in A^j$ and $a_{ij}=0$ for all $i\notin A^j$. The third inequality holds because $\gamma_j^1 \geq \gamma_j^2 \geq \ldots \geq \gamma_j^T \geq \gamma_j^{T+1}$. The second equality follows by (15). The fourth inequality follows by the fact that $M = \max_{i\in\mathcal{L}} m_i$ and $|A^k|\leq L$ for all $k\in\mathcal{J}$. Therefore, we get $\mathsf{Z}^*_{LP} \leq (1+\theta\,(2M-1)L)\sum_{j\in\mathcal{J}}\gamma_j^1$. Since the optimal objective value of the linear program in (23) is an upper bound on the optimal total expected revenue, we obtain $V^1(\boldsymbol{C}) \leq \mathsf{Z}^*_{LP} \leq (1+\theta\,(2M-1)L)\sum_{j\in\mathcal{J}}\gamma_j^1$. □

As discussed at the beginning of this appendix, if $r_j \geq V^{t+1}(\boldsymbol{x}) - V^{t+1}(\boldsymbol{x} - \sum_{i\in A^j} a_{ij}\boldsymbol{e}_i)$, then it is optimal to accept a request for product j, as long as we have $\prod_{i\in A^j} \mathbb{1}_{\{x_i\geq a_{ij}\}} = 1$. Replacing

V^{t+1} in the last inequality with the value function approximation H^{t+1}, we obtain our approximate policy. So, the decision function of the approximate policy is given by

$$u_j^{\mathsf{App},t}(\boldsymbol{x}) = \begin{cases} \prod_{i \in A^j} \mathbb{1}_{\{x_i \geq a_{ij}\}} & \text{if } \; r_j \geq H^{t+1}(\boldsymbol{x}) - H^{t+1}\left(\boldsymbol{x} - \sum_{i \in A^j} a_{ij}\, \boldsymbol{e}_i\right) \\ 0 & \text{otherwise.} \end{cases} \tag{24}$$

Letting $U^t(\boldsymbol{x})$ be the total expected revenue of the approximate policy over time periods $t, t+1, \ldots, T$ given that the system is in state $\boldsymbol{x}$ at time period t, we have the recursion

$$\begin{aligned} U^t(\boldsymbol{x}) &= \sum_{j \in \mathcal{J}} \lambda_j^t\, u_j^{\mathsf{App},t}(\boldsymbol{x}) \left[r_j + U^{t+1}\left(\boldsymbol{x} - \sum_{i \in A^j} a_{ij}\, \boldsymbol{e}_i\right)\right] + \left(1 - \sum_{j \in \mathcal{J}} \lambda_j^t + \sum_{j \in \mathcal{J}} \lambda_j^t\, (1 - u_j^{\mathsf{App},t}(\boldsymbol{x}))\right) U^{t+1}(\boldsymbol{x}) \\ &= U^{t+1}(\boldsymbol{x}) + \sum_{j \in \mathcal{J}} \lambda_j^t\, u_j^{\mathsf{App},t}(\boldsymbol{x}) \left[r_j - U^{t+1}(\boldsymbol{x}) + U^{t+1}\left(\boldsymbol{x} - \sum_{i \in A^j} a_{ij}\, \boldsymbol{e}_i\right)\right], \end{aligned} \tag{25}$$

with the boundary condition that $U^{T+1} = 0$. The recursion above is the analogue of the recursion in (8) for the case where each product can consume multiple units of a resource.

The next lemma gives a lower bound on the total expected revenue obtained by the approximate policy. It generalizes Lemma 3.8 to multiple units of resource consumption by the products.

Lemma C.3 *If the tuning parameter θ satisfies $\theta \geq \Delta_{\mathscr{B}}$ and the coefficients $\{\gamma_j^t : j \in \mathcal{J},\; t \in \mathcal{T}\}$ are computed through* (15)*, then we have $U^1(\boldsymbol{C}) \geq \sum_{j \in \mathcal{J}} \gamma_j^1$.*

Proof: We use induction over the time periods to show that $U^t(\boldsymbol{x}) \geq H^t(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{Q}$ and $t \in \mathcal{T}$, where $H^t(\boldsymbol{x}) = \sum_{j \in \mathcal{J}} \gamma_j^t\, \varphi_j(\boldsymbol{x})$ with $\{\gamma_j^t : j \in \mathcal{J},\; t \in \mathcal{T}\}$ computed through (15). Consider the base case at time period $T+1$. Since $U^{T+1} = 0 = H^{T+1}$, the base case holds. Suppose that that the result holds at time period $t+1$ so that $U^{t+1}(\boldsymbol{x}) \geq H^{t+1}(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{Q}$. Replacing U^{t+1} on the right side of (25) with H^{t+1}, we obtain

$$\begin{aligned} U^t(\boldsymbol{x}) &\geq H^{t+1}(\boldsymbol{x}) + \sum_{j \in \mathcal{J}} \lambda_j^t\, u_j^{\mathsf{App},t}(\boldsymbol{x}) \left[r_j - H^{t+1}(\boldsymbol{x}) + H^{t+1}\left(\boldsymbol{x} - \sum_{i \in A^j} a_{ij}\, \boldsymbol{e}_i\right)\right] \\ &= H^{t+1}(\boldsymbol{x}) + \sum_{j \in \mathcal{J}} \lambda_j^t \left(\prod_{i \in A^j} \mathbb{1}_{\{x_i \geq a_{ij}\}}\right) \left[r_j - H^{t+1}(\boldsymbol{x}) + H^{t+1}\left(\boldsymbol{x} - \sum_{i \in A^j} a_{ij}\, \boldsymbol{e}_i\right)\right]^+ \\ &\geq H^{t+1}(\boldsymbol{x}) + \sum_{j \in \mathcal{J}} \lambda_j^t \left(\prod_{i \in A^j} \mathbb{1}_{\{x_i \geq a_{ij}\}}\right) \left[r_j - \theta \sum_{i \in A^j} \frac{2m_i - 1}{C_i} \sum_{k \in \mathcal{J}} \mathbb{1}_{\{i \in A^k\}} \gamma_k^{t+1}\right]^+ \\ &= H^{t+1}(\boldsymbol{x}) + \sum_{j \in \mathcal{J}} \left(\prod_{i \in A^j} \mathbb{1}_{\{x_i \geq a_{ij}\}}\right) \left(\gamma_j^t - \gamma_j^{t+1}\right) \;\geq\; H^{t+1}(\boldsymbol{x}) + \sum_{j \in \mathcal{J}} \varphi_j(\mathtt{G}_j(\boldsymbol{x})) \left(\gamma_j^t - \gamma_j^{t+1}\right) \;=\; H^t(\boldsymbol{x}). \end{aligned}$$

In the chain of inequalities above, the first equality holds by the definition of the decision function for the approximate policy given in (24). The second inequality follows by noting that

$H^t(\boldsymbol{x}) - H^t(\boldsymbol{x} - \sum_{i\in A^j} a_{ij}\, \boldsymbol{e}_i) \;\le\; \Delta_{\mathscr{B}} \sum_{i\in A^j} \frac{2m_i-1}{C_i} \sum_{k\in\mathcal{J}} \mathbb{1}_{\{i\in A^k\}}\, \gamma_k^t$ by Lemma C.1 and we choose the tuning parameter θ to satisfy $\theta \ge \Delta_{\mathscr{B}}$. The second equality follows from (15). To see that the third inequality holds, note that we have $\varphi_j(\mathtt{G}_j(\boldsymbol{x})) \;\le\; \prod_{i\in A^j} \mathbb{1}_{\{x_i\ge a_{ij}\}}$. In particular, if $\prod_{i\in A^j} \mathbb{1}_{\{x_i\ge a_{ij}\}} = 1$, then the last inequality holds trivially since $\varphi_j : \mathcal{Q} \to [0,1]$ for all $j\in\mathcal{J}$. If, on the other hand, $\prod_{i\in A^j} \mathbb{1}_{\{x_i\ge a_{ij}\}} = 0$, then there must exist some $i\in A^j$ such that $x_i < a_{ij}$, in which case, the i-th component of $\mathtt{G}_j(\boldsymbol{x})$ will be zero by the definition of $\mathtt{G}_j$, so by part (a) of Definition 3.1, we get $\varphi_j(\mathtt{G}_j(\boldsymbol{x})) = 0$. Therefore, the fourth inequality holds because we have $\varphi_j(\mathtt{G}_j(\boldsymbol{x})) \;\le\; \prod_{i\in A^j} \mathbb{1}_{\{x_i\ge a_{ij}\}}$ and (15) implies that $\gamma_j^t - \gamma_j^{t+1} \ge 0$. The last equality holds because $H^{t+1}(\boldsymbol{x}) = \sum_{j\in\mathcal{J}} \gamma_j^{t+1}\, \varphi_j(\mathtt{G}_j(\boldsymbol{x}))$. The chain of inequalities above completes the induction argument, so it follows that $U^t(\boldsymbol{x}) \ge H^t(\boldsymbol{x})$ for all $\boldsymbol{x}\in\mathcal{Q}$ and $t\in\mathcal{T}$. Using the last inequality with $t=1$ and $\boldsymbol{x} = \boldsymbol{C}$, we get $U^1(\boldsymbol{C}) \ge H^1(\boldsymbol{C}) = \sum_{j\in\mathcal{J}} \gamma_j^1\, \varphi_j(\mathtt{G}_j(\boldsymbol{C})) = \sum_{j\in\mathcal{J}} \gamma_j^1\, \varphi_j(\boldsymbol{C}) = \sum_{j\in\mathcal{J}} \gamma_j^1$. $\square$

The proof of Theorem 4.3 follows by combining Lemmas C.2 and C.3.

Proof of Theorem 4.3: By Lemma C.2, we have $(1+\theta(2M-1)L) \sum_{j\in\mathcal{J}} \gamma_j^1 \ge V^1(\boldsymbol{C})$, whereas by Lemma C.3, we have $U^1(\boldsymbol{C}) \ge \sum_{j\in\mathcal{J}} \gamma_j^1$. So, $U^1(\boldsymbol{C}) \ge \sum_{j\in\mathcal{J}} \gamma_j^1 \ge V^1(\boldsymbol{C})/(1+\theta\,(2M-1)L)$. $\square$

Appendix D: Performance Guarantee by Leveraging a Linear Programming Approximation

In this section, we give a proof for Theorem 4.4. In the next lemma, we give an upper bound on the optimal objective value of the linear program in (4).

Lemma D.1 *Noting that Z_{LP}^* is the optimal objective value of the linear program in* (4), *if the coefficients $\{\gamma_j^t : j\in\mathcal{J},\ t\in\mathcal{T}\}$ are computed through* (16), *then we have $\mathsf{Z}_{LP}^* \le (1+\theta L) \sum_{j\in\mathcal{J}} \gamma_j^1$.*

Proof: Recall that Λ_j in the linear program in (4) is given by $\Lambda_j = \sum_{t\in\mathcal{T}} \lambda_j^t$. In this case, since we have $\gamma_j^{T+1} = 0$ for all $j\in\mathcal{J}$, a telescoping sum yields

$$\begin{aligned}
\sum_{j\in\mathcal{J}} \gamma_j^1 \;&=\; \sum_{j\in\mathcal{J}}\sum_{t\in\mathcal{T}} (\gamma_j^t - \gamma_j^{t+1}) \;=\; \sum_{t\in\mathcal{T}}\sum_{j\in\mathcal{J}} \frac{z_j^*}{\Lambda_j}\lambda_j^t \left[r_j - \theta \sum_{i\in A^j} \frac{1}{C_i} \sum_{k\in\mathcal{J}} \mathbb{1}_{\{i\in A^k\}} \gamma_k^{t+1} \right]^+ \\
&\ge\; \sum_{t\in\mathcal{T}}\sum_{j\in\mathcal{J}} \frac{z_j^*}{\Lambda_j}\lambda_j^t \left[r_j - \theta \sum_{i\in A^j} \frac{1}{C_i} \sum_{k\in\mathcal{J}} \mathbb{1}_{\{i\in A^k\}} \gamma_k^{t+1} \right] \;=\; \mathsf{Z}_{LP}^* - \sum_{t\in\mathcal{T}}\sum_{j\in\mathcal{J}} \theta\, \frac{z_j^*}{\Lambda_j}\, \lambda_j^t \sum_{i\in A^j} \frac{1}{C_i} \sum_{k\in\mathcal{J}} \mathbb{1}_{\{i\in A^k\}}\, \gamma_k^{t+1} \\
&\ge\; \mathsf{Z}_{LP}^* - \sum_{t\in\mathcal{T}}\sum_{j\in\mathcal{J}} \theta\, \frac{z_j^*}{\Lambda_j}\, \lambda_j^t \sum_{i\in A^j} \frac{1}{C_i} \sum_{k\in\mathcal{J}} \mathbb{1}_{\{i\in A^k\}}\, \gamma_k^1 \;=\; \mathsf{Z}_{LP}^* - \sum_{j\in\mathcal{J}} \theta\, z_j^* \sum_{i\in\mathcal{L}} \mathbb{1}_{\{i\in A^j\}} \frac{1}{C_i} \sum_{k\in\mathcal{J}} \mathbb{1}_{\{i\in A^k\}}\, \gamma_k^1 \\
&=\; \mathsf{Z}_{LP}^* - \theta \sum_{i\in\mathcal{L}}\sum_{k\in\mathcal{J}} \mathbb{1}_{\{i\in A^k\}}\, \gamma_k^1\, \frac{1}{C_i} \sum_{j\in\mathcal{J}} \mathbb{1}_{\{i\in A^j\}}\, z_j^* \;\ge\; \mathsf{Z}_{LP}^* - \theta \sum_{i\in\mathcal{L}}\sum_{k\in\mathcal{J}} \mathbb{1}_{\{i\in A^k\}}\, \gamma_k^1 \;\ge\; \mathsf{Z}_{LP}^* - \theta L \sum_{k\in\mathcal{J}} \gamma_k^1.
\end{aligned}$$

In the chain of inequalities above, the second equality follows from (16). The third equality follows because $\sum_{t\in\mathcal{T}} \lambda_j^t = \Lambda_j$ and $\mathsf{Z}_{LP}^* = \sum_{j\in\mathcal{J}} r_j\, z_j^*$. The second inequality holds because

$\gamma_j^1 \geq \gamma_j^2 \geq \ldots \geq \gamma_j^{T+1}$ by (16). The fourth equality uses the fact that $i \in A^j$ if and only if $\mathbb{1}_{\{i\in A^j\}} = 1$. The third inequality holds because the first constraint in problem (4) implies that $\sum_{j\in\mathcal{J}} \mathbb{1}_{\{i\in A^j\}}\, z_j^* \leq C_i$. The fourth inequality uses the fact that $\sum_{i\in A^k} \mathbb{1}_{\{i\in A^k\}} = |A^k| \leq L$. By the chain of inequalities, we have $(1+\theta L)\sum_{j\in\mathcal{J}} \gamma_j^1 \geq \mathsf{Z}_{LP}^*$, as desired. $\square$

The decision function of our approximate policy has the same form as (5). The only difference is that we compute the coefficients $\{\gamma_j^t : j\in\mathcal{J},\ t\in\mathcal{T}\}$ by using (16). To compute the total expected revenue obtained by our approximate policy, we can use the same recursion in (8). Next, we give a lower bound on the total expected revenue obtained by our approximate policy.

Lemma D.2 *If the tuning parameter θ satisfies $\theta \geq \Delta_{\mathscr{B}}$ and the coefficients $\{\gamma_j^t : j\in\mathcal{J},\ t\in\mathcal{T}\}$ are computed through* (16)*, then we have $U^1(\boldsymbol{C}) \geq \sum_{j\in\mathcal{J}} \gamma_j^1$.*

Proof: We will use induction over the time periods to show that $U^t(\boldsymbol{x}) \geq H^t(\boldsymbol{x})$ for all $\boldsymbol{x}\in\mathcal{Q}$ and $t\in\mathcal{T}$. Using this inequality with $t=1$ and $\boldsymbol{x}=\boldsymbol{C}$ yields $U^1(\boldsymbol{C}) \geq H^1(\boldsymbol{C}) = \sum_{j\in\mathcal{J}}\gamma_j^1$, as desired. The base case at $t=T+1$ holds because $U^{T+1}=0=H^{T+1}$. Suppose that $U^{t+1}\geq H^{t+1}$. Replacing U^{t+1} on the right side of (8) with H^{t+1}, the right side of (8) becomes smaller, so we get

$$
\begin{aligned}
U^t(\boldsymbol{x}) \;&\geq\; H^{t+1}(\boldsymbol{x}) + \sum_{j\in\mathcal{J}} \lambda_j^t\, u_j^{\mathsf{App},t}(\boldsymbol{x}) \left[r_j - H^{t+1}(\boldsymbol{x}) + H^{t+1}\left(\boldsymbol{x} - \sum_{i\in A^j} \boldsymbol{e}_i\right)\right] \\
&=\; H^{t+1}(\boldsymbol{x}) + \sum_{j\in\mathcal{J}} \lambda_j^t \left(\prod_{i\in A^j} \mathbb{1}_{\{x_i\geq 1\}}\right) \left[r_j - H^{t+1}(\boldsymbol{x}) + H^{t+1}\left(\boldsymbol{x} - \sum_{i\in A^j} \boldsymbol{e}_i\right)\right]^+ \\
&\geq\; H^{t+1}(\boldsymbol{x}) + \sum_{j\in\mathcal{J}} \lambda_j^t \left(\prod_{i\in A^j} \mathbb{1}_{\{x_i\geq 1\}}\right) \left[r_j - \Delta_{\mathscr{B}} \sum_{i\in A^j} \frac{1}{C_i} \sum_{k\in\mathcal{J}} \mathbb{1}_{\{i\in A^k\}} \gamma_k^{t+1}\right]^+ \\
&\geq\; H^{t+1}(\boldsymbol{x}) + \sum_{j\in\mathcal{J}} \frac{z_j^*}{\Lambda_j}\,\lambda_j^t \left(\prod_{i\in A^j} \mathbb{1}_{\{x_i\geq 1\}}\right) \left[r_j - \theta \sum_{i\in A^j} \frac{1}{C_i} \sum_{k\in\mathcal{J}} \mathbb{1}_{\{i\in A^k\}} \gamma_k^{t+1}\right]^+ \\
&=\; H^{t+1}(\boldsymbol{x}) + \sum_{j\in\mathcal{J}} \left(\prod_{i\in A^j} \mathbb{1}_{\{x_i\geq 1\}}\right) \left(\gamma_j^t - \gamma_j^{t+1}\right) \;\geq\; H^{t+1}(\boldsymbol{x}) + \sum_{j\in\mathcal{J}} \varphi_j(\boldsymbol{x}) \left(\gamma_j^t - \gamma_j^{t+1}\right) \;=\; H^t(\boldsymbol{x}),
\end{aligned}
$$

where we use the same argument used in the chain of inequalities in the proof of Lemma 3.8 in the main text. The only difference is that the third inequality uses the fact that $z_j^* \leq \Lambda_j$ by the second constraint in problem (4), along with the fact that $\theta \geq \Delta_{\mathscr{B}}$. The chain of inequalities above completes the induction argument so that $U^t(\boldsymbol{x}) \geq H^t(\boldsymbol{x})$ for all $\boldsymbol{x}\in\mathcal{Q}$ and $t\in\mathcal{T}$. $\square$

Here is the proof of Theorem 4.4.

Proof of Theorem 4.4: The optimal objective value of the linear program in (4) is an upper bound on the optimal total expected revenue, so we have $\mathsf{Z}_{LP}^* \geq V^1(\boldsymbol{C})$; see Bertsimas and Popescu (2003). In this case, using Lemmas D.1 and D.2, it follows that $U^1(\boldsymbol{C}) \geq \sum_{j\in\mathcal{J}}\gamma_j^1 \geq \mathsf{Z}_{LP}^*/(1+\theta L) \geq V^1(\boldsymbol{C})/(1+\theta L)$. $\square$

Appendix E: Performance of the Online Packing Policy

We give a simple example to show that the policy proposed by Kesselheim et al. (2014) can perform arbitrarily poorly under nonstationary arrivals. We briefly describe this policy. Given that we are at time period t, we use ζ_j^{t-1} to denote the total number of requests that we received for product j over time periods $1,\ldots,t-1$. Using the decision variables $\boldsymbol{z} = (z_j : j \in \mathcal{J})$ and y, if we have a request for product k at time period t, then we solve the linear program

$$\max\left\{ \sum_{j\in\mathcal{J}} r_j\, z_j + r_k\, y \;:\; \sum_{j\in\mathcal{J}} \mathbb{1}_{\{i\in A^j\}}\, z_j + \mathbb{1}_{\{i\in A^k\}}\, y \le \frac{t}{T}\, C_i \quad \forall\, i\in\mathcal{L},\right.$$
$$\left. 0\le z_j \le \zeta_j^{t-1} \quad \forall\, j\in\mathcal{J}, \quad 0\le y\le 1 \right\}. \qquad (26)$$

We can view the problem above as a version of the linear program in (4) in which we only focus on the product requests that have actually been realized over time periods $1,\ldots,t$ and we scale the capacities of the resources by t/T. In this case, letting $(\boldsymbol{z}^*, y^*)$ be an optimal solution to the linear program above, we accept the request for product k at time period t with probability y^*. This discussion specifies the policy proposed by Kesselheim et al. (2014).

The linear program above only uses the actual realizations of the product requests. In particular, it does not require forecasting. Not so surprisingly, this feature becomes problematic when the arrivals of the product requests are nonstationary. We proceed to give an example where the policy proposed by Kesselheim et al. (2014) can perform arbitrarily poorly under nonstationary arrivals. Consider a problem instance with a single resource with a capacity of 1 unit. There is a single product with a revenue of 1. The product uses the single resource. There are T time periods in the selling horizon. At the first time period, we have a request for the product with probability 1. At each of the remaining $T-1$ time periods, we do not have any product requests.

For this problem instance, consider problem (26) at the first time period. Noting that there is a request for the product with probability one at the first time period, problem (26) takes the form $\max\{y \,:\, y\le 1/T,\;\; 0\le y\le 1\}$, yielding the optimal solution $y^* = 1/T$. So, we accept the request at the first time period with probability $1/T$. Since there are no product requests at the other time periods, the total expected revenue obtained by the policy proposed by Kesselheim et al. (2014) is $1/T$. The optimal policy always accepts the product request at the first time period, yielding a total expected revenue of 1. If we choose T arbitrarily large, then the policy proposed by Kesselheim et al. (2014) performs arbitrarily poorly when compared with the optimal policy.

The undesirable performance of the policy proposed by Kesselheim et al. (2014) is due to the nonstationary arrivals. For this simple problem instance, we note that our approximate policy always accepts the request at the first time period, yielding the optimal total expected revenue.

Appendix F: Computational Experiments for Hotel Network Revenue Management

We provide computational experiments in the hotel network revenue management setting, where we systematically vary the maximum number of resources used by a product.

F.1 Experimental Setup and Benchmark Methods

We consider the booking process of a hotel with a single room type over τ days. In all of our test problems, we fix the number of days as $\tau = 60$. Each resource corresponds to the room capacity on a particular day. Therefore, there are τ resources, which we index by $\mathcal{L} = \{1, \ldots, \tau\}$. A customer can book a room for up to L contiguous days and at κ different fare levels. A product request corresponds to a customer interested in booking a room for a particular number of days, beginning on a particular day and at a particular fare level. Therefore, there are $L \times \tau \times \kappa$ products, indexed by $\mathcal{J} = \{(\ell, s, k) : \ell = 1, \ldots, L,\ s = 1, \ldots, \tau,\ k = 1, \ldots, \kappa\}$. A customer with a request for product (ℓ, s, k) is interested in staying for ℓ days, beginning on day s and by paying the fare level k. In this case, product (ℓ, s, k) uses the capacities of the resources $s, s+1, \ldots, \min\{s+\ell-1, \tau\}$, so the maximum number of resources used by a product is L. In our computational experiments, we fix $\kappa = 2$ so that there is only a high-fare and a low-fare. We work with different values of L, allowing us to systematically vary the maximum number of resources used by a product.

If a low-fare product corresponds to staying for ℓ days, then its revenue is $10\,\ell\,(1 - \ell/2L)$. Thus, if a low-fare customer stays for ℓ days, then she pays $10\,(1 - \ell/2L)$ per day, which implies that the customers staying longer pay less per day. The revenue from a high-fare product is four times the revenue from the corresponding low-fare product. Bookings for staying on a particular day start arriving 15 days before the beginning day of the stay. Therefore, the requests for product (ℓ, s, k) arrive over the days $\max\{s - 15, 1\}, \ldots, s$. Over these days, the requests for the low-fare products tend to arrive earlier, whereas the requests for the high-fare products tend to arrive later. To generate test problems with this nature, we used an approach that is very similar to the one used for the airline network revenue management test problems. The arrival probability for a low-fare product decreases as we get closer to the day of stay for the product, whereas the arrival probability for a high-fare product increases as we get closer to the day of stay for the product. The length of stay for a product request is equally likely to take any of the values $\{1, \ldots, L\}$.

We divide each day into 30 time periods so that there is one product request at each time period. Thus, the number of time periods in the selling horizon is $T = 60 \times 30 = 1,800$. Using $\lambda^t_{(\ell,s,k)}$ to denote the probability that we have a request for product (ℓ, s, k) at time period t, since product (ℓ, s, k) uses the capacities of the resources $s, s+1, \ldots, \min\{s+\ell-1, \tau\}$, the total expected demand for the capacity on day i is $\sum_{t \in \mathcal{T}} \sum_{(\ell,s,k) \in \mathcal{J}} \mathbb{1}_{\{s \le i \le \min\{s+\ell-1,\tau\}\}}\, \lambda^t_{(\ell,s,k)}$. We set the capacity of resource

i as $C_i = \frac{1}{\alpha} \sum_{t \in \mathcal{T}} \sum_{(\ell,s,k) \in \mathcal{J}} \mathbb{1}_{\{s \le i \le \min\{s+\ell-1,\tau\}\}} \lambda^t_{(\ell,s,k)}$, so larger values for α yield tighter capacities. We vary α in our computational experiments. Varying $L \in \{1,2,3,4,5\}$ and $\alpha \in \{1.0, 1.2, 1.6\}$, we obtain 15 test problems in our experimental setup. With these values for L and α, in our test problems, the room capacities per day range between 59 to 94. We use the same benchmarks that we used for the airline network revenue management test problems. In particular, APP refers to our approximate policy with the basis function $\varphi_j(\boldsymbol{x}) = \min_{i \in A^j} (1 - e^{-x_i/C_i})/(1-e^{-1})$, BPP refers to the bid price policy, RLP refers to the randomized linear program, DIF refers to the finite differences, DEC refers to the dynamic programming decomposition, and OPP refers to the online packing policy proposed by Kesselheim et al. (2014).

F.2 Computational Results

We give our computational results in Table 5. The layout of this table is similar to that of Tables 1 and 2. The only difference is that we use the tuple (L, α) to capture the parameter configuration of each test problem. Our results for hotel network revenue management closely reflect those for airline network revenue management. The performance of APP is better than that of BPP, RLP and DIFF with average total expected revenue gaps of 7.88%, 3.80% and 5.85%, respectively, over all of our test problems. APP lags behind DEC. As mentioned earlier, DEC is one of the strongest heuristics for network revenue management problems in practice, but it does not have a performance guarantee. The average gap between the performance of APP and DEC is -1.18%. The performance of OPP is not competitive.

The performance guarantee of $1/(1+\theta L)$ for APP deteriorates as L gets larger so that the maximum number of resources used by a product increases. On the one hand, this deterioration in the theoretical guarantee is not reflected in our computational results when we measure the gap between the upper bound on the optimal total expected revenue and the total expected revenue obtained by APP. Although these figures are not reported in Table 5, if we consider the test problems with $L = 1$, $L = 2$, $L = 3$, $L = 4$ and $L = 5$ separately, then the average gaps between the upper bound on the optimal total expected revenue and the total expected revenue obtained by APP are, respectively, 8.54%, 8.86%, 8.33%, 7.58% and 6.28%. Thus, the performance of APP, relative to the upper bound on the optimal total expected revenue, is robust as L gets larger.

On the other hand, the performance gaps between APP and the other benchmarks indeed decrease as L gets larger. In Table 6, we show the average percent gap between the total expected revenues obtained by APP and the other benchmarks, where the average is computed over all test problems with a particular value of L. For example, the top left figure in the table shows the average percent gap between the performance of APP and BPP for the test problems $(1, 1.0)$,

Params. (L,α)	Upp. Bnd.	Tot. Exp. Rev. APP	BPP	RLP	DIF	DEC	OPP	% Gap with APP BPP	RLP	DIF	DEC	OPP
(1,1.0)	9,802	8,961	8,564	8,714	8,596	9,060	7,934	4.43	2.76	4.07	-1.10	11.47
(1,1.2)	9,124	8,349	7,365	7,812	7,461	8,431	6,770	11.78	6.43	10.63	-0.98	18.91
(1,1.6)	8,060	7,372	5,724	6,866	6,021	7,405	4,842	22.35	6.86	18.33	-0.45	34.32
(2,1.0)	11,127	10,216	9,773	9,902	9,846	10,338	9,056	4.34	3.08	3.62	-1.20	11.35
(2,1.2)	10,352	9,480	8,408	8,832	8,586	9,575	7,624	11.31	6.83	9.43	-1.01	19.58
(2,1.6)	9,370	8,438	7,400	8,150	7,747	8,573	5,391	12.30	3.41	8.19	-1.60	36.11
(3,1.0)	13,723	12,729	12,171	12,275	12,282	12,845	11,292	4.38	3.56	3.50	-0.92	11.28
(3,1.2)	12,942	11,930	10,879	11,189	11,300	12,041	9,585	8.81	6.21	5.28	-0.93	19.65
(3,1.6)	11,607	10,456	9,667	10,116	9,910	10,704	6,757	7.55	3.25	5.22	-2.38	35.38
(4,1.0)	17,503	16,300	15,717	15,931	15,807	16,443	14,507	3.58	2.26	3.03	-0.87	11.00
(4,1.2)	16,583	15,356	14,262	14,662	14,755	15,520	12,262	7.12	4.52	3.91	-1.07	20.15
(4,1.6)	14,870	13,610	12,762	13,335	12,973	13,807	8,718	6.22	2.02	4.68	-1.45	35.94
(5,1.0)	24,128	22,730	22,162	22,410	22,376	22,959	20,399	2.50	1.41	1.56	-1.00	10.26
(5,1.2)	22,828	21,487	20,052	20,761	20,720	21,753	17,149	6.68	3.38	3.57	-1.23	20.19
(5,1.6)	20,644	19,164	18,239	18,955	18,643	19,447	11,969	4.82	1.09	2.72	-1.48	37.55
Average								7.88	3.80	5.85	-1.18	22.21

Table 5 Computational results for the hotel network revenue management test problems.

L	% Gap with APP BPP	RLP	DIF	DEC	OPP
1	12.86	5.35	11.01	-0.85	21.57
2	9.32	4.44	7.08	-1.27	22.35
3	6.91	4.34	4.67	-1.41	22.11
4	5.64	2.93	3.87	-1.13	22.36
5	4.67	1.96	2.62	-1.24	22.66

Table 6 Average performance gaps between APP and the other benchmarks as a function of L.

$(1,1.2)$ and $(1,1.6)$. Our results indicate that as L gets larger, the performance gaps between APP and the other benchmarks decrease, except for OPP. Naturally, without knowing the optimal total expected revenue, it is difficult to distinguish whether this deterioration is due to the fact that APP performs worse or the other benchmarks perform better as L gets larger.

Lastly, we experimented with APP using each one of the basis functions in Table 4. For all of our test problems, the performance of APP was relatively insensitive to the choice of the basis functions, but the basis function $\varphi_j(\boldsymbol{x}) = \min_{i \in A^j}(1 - e^{-x_i/C_i})/(1 - e^{-1})$, which is the basis function that we use, provided small improvements over the others.

Appendix G: Computational Experiments under Stationary Arrivals

In this section, we provide computational experiments to test the performance of OPP under stationary arrivals. Our experimental setup is based on the airline network revenue management test problems that we use in Section 5. We generate our test problems by using the same approach discussed in Section 5.1. The only difference is in the way we generate the arrival probabilities for the product requests, as we focus on stationary arrivals. In particular, after generating $B_{o,d}$ as in Section 5.1, if product j is the high-fare or low-fare itinerary for origin-destination pair (o,d), then

Params. (T, N, κ, α)	Upp. Bnd.	Tot. Exp. Rev. APP	OPP	% Gap
$(200, 6, 4, 1.0)$	30,990	29,241	28,049	4.1
$(200, 6, 4, 1.2)$	29,562	27,687	26,174	5.5
$(200, 6, 4, 1.6)$	27,146	24,486	22,702	7.3
$(200, 6, 8, 1.0)$	55,820	53,491	51,393	3.9
$(200, 6, 8, 1.2)$	54,376	51,685	48,656	5.9
$(200, 6, 8, 1.6)$	51,950	48,280	43,356	10.2
Average				6.1

Params. (T, N, κ, α)	Upp. Bnd.	Tot. Exp. Rev. APP	OPP	% Gap
$(600, 6, 4, 1.0)$	93,083	90,481	89,023	1.6
$(600, 6, 4, 1.2)$	88,716	86,268	83,809	2.9
$(600, 6, 4, 1.6)$	81,141	75,201	74,152	1.4
$(600, 6, 8, 1.0)$	167,575	164,363	161,901	1.5
$(600, 6, 8, 1.2)$	163,149	159,728	155,033	2.9
$(600, 6, 8, 1.6)$	155,553	151,136	142,115	6.0
Average				2.7

Table 7 Computational results for the test problems with stationary arrivals.

we set $\lambda_j^t = \frac{1}{2} B_{o,d} / \sum_{(s,r)\in\mathcal{D}} B_{s,r}$ for all $t \in \mathcal{T}$. For economy of space, we focus on test problems with six spokes and compare the performance of APP and OPP. We give our results in Table 7. In the first column, we show the parameter configuration for each test problem using the tuple (T, N, κ, α), where N, κ and α are as discussed in Section 5.1. In the second column, we show the upper bound on the optimal total expected revenue provided by the optimal objective value of problem (4). In the third and fourth columns, we show the total expected revenues obtained by APP and OPP. In the fifth column, we show the percent gap between the total expected revenues obtained APP and OPP. The results in Table 7 indicate that APP continues to provide significant improvements over OPP under stationary arrivals. Comparing the performance gaps between APP and OPP in Table 7 with those in Tables 1 and 2, the performance of OPP relative to APP improves under stationary arrivals, but this improvement is not enough to catch up with APP.

Appendix H: Computational Results without Recomputing Policy Parameters

In this section, we focus on a portion of the test problems in Section 5 and test the performance of the benchmarks when we compute the policy parameters only once at the beginning of the selling horizon. We give our results in Table 8. The layout of this table is identical to that of Table 1. We do not provide comparisons with OPP since OPP, by its definition, has to solve a linear program at each time period. Comparing the results in Table 8 with those in Table 1, we observe that the performance of APP deteriorates slightly when we do not recompute the policy parameters. The performance of DEC remains essentially the same. The performance of all other benchmarks deteriorates substantially when we do not recompute the policy parameters.

References

Bertsimas, D., I. Popescu. 2003. Revenue management in a dynamic network environment. *Transportation Science* **37**(3) 257–277.

Kesselheim, T., A. Tönnis, K. Radke, B. Vöcking. 2014. Primal beats dual on online packing LPs in the random-order model. *Proceedings of the Forty-sixth Annual ACM Symposium on Theory of Computing*. STOC '14, New York, NY, 303–312.

Params. (T, N, κ, α)	Upp. Bnd.	Tot. Exp. Rev. APP	BPP	RLP	DIF	DEC	% Gap with APP BPP	RLP	DIF	DEC
$(200, 6, 4, 1.0)$	22,300	20,565	18,285	18,829	18,395	20,764	11.09	8.44	10.55	-0.97
$(200, 6, 4, 1.2)$	20,932	18,959	15,290	15,430	17,160	19,271	19.35	18.61	9.49	-1.64
$(200, 6, 4, 1.6)$	18,592	16,275	12,613	13,194	13,052	16,834	22.50	18.93	19.81	-3.44
$(200, 6, 8, 1.0)$	35,544	33,545	27,682	28,712	27,907	33,733	17.48	14.41	16.81	-0.56
$(200, 6, 8, 1.2)$	34,172	31,808	22,315	22,430	23,789	32,154	29.85	29.48	25.21	-1.09
$(200, 6, 8, 1.6)$	31,824	29,231	18,840	20,229	20,942	29,625	35.55	30.79	28.36	-1.35
Average							22.64	20.11	18.37	-1.51

Table 8 Computational results without recomputing the policy parameters.

Liu, Q., G. J. van Ryzin. 2008. On the choice-based linear programming model for network revenue management. *Manufacturing & Service Operations Management* **10**(2) 288–310.