A Constant-Factor Approximation Algorithm for Network Revenue Management

Yuhang Ma\textsuperscript{1}, Paat Rusmevichientong\textsuperscript{2}, Mika Sumida\textsuperscript{1}, Huseyin Topaloglu\textsuperscript{1}

\textsuperscript{1} School of Operations Research and Information Engineering, Cornell Tech, New York, NY 10044, 
\textsuperscript{2} Marshall School of Business, University of Southern California, Los Angeles, CA 90089, 
ym367@cornell.edu, rusmevic@marshall.usc.edu, ms3268@cornell.edu, ht88@cornell.edu

April 8, 2018

We provide a constant-factor 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. To our knowledge, our approximate policy is the first constant-factor approximation algorithm for network revenue management problems. 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.

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 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 a constant-factor approximation algorithm for network revenue management problems. Our problem setup follows the standard network revenue management literature. We have access to 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 enters the system with a request for a particular product. If we accept the product request, then we generate a certain amount of revenue and consume capacities of the different resources used by the product. The goal is to find a policy to determine which product requests to serve 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 capacities of all the resources. Therefore, it is intractable to compute the optimal policy even when we have a relatively small number of resources.

**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 constant-factor approximation algorithm for network revenue management problems. Moreover, 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 of 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 in the selling horizon to compute coefficients for the basis functions
in the linear combinations. In our backward recursion, we have a tuning parameter \( \theta \) 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 \( \theta \) gets smaller. In our approach,
the tuning parameter \( \theta \) 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 \( \theta \), in which case,
we obtain the performance guarantee of \( 1/(1 + L) \) in the previous paragraph.

Second, we start with a network revenue management setup where each customer enters 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 how to extend our approach to a case in which
we offer a set of products to each arriving customer, and the customer chooses among the offered
products. Similarly, we show how to extend our approach to a case in which each product uses
more than one unit of a resource, which happens to be the case in airlines, for example, 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 consumes
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 incorporate the solution to a linear programming
approximation, while retaining the performance guarantee of \( 1/(1 + L) \).

Our computational experiments demonstrate that the approximate policy that we provide
performs substantially better than its theoretical performance guarantee, when compared with a
tractable upper bound on the optimal total expected revenue. We also compare our approximate
policy with the standard bid price policy from a linear programming approximation and
demonstrate that our approximate policy performs noticeably better. We emphasize that the
attractive feature of our approximate policy is that it provides a constant-factor performance
guarantee, and it is the first policy in the literature to possess this feature. As we discuss below in
our literature review, there are heuristic policies to address various shortcomings of the standard bid
price policy, but none of these policies have constant-factor approximation guarantees. The question of whether we can further improve the theoretical or practical performance of our approximate policy is certainly an interesting research question to pursue.

**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 capacity. In this case, we are willing to accept a product request if the revenue from the product request exceeds the total value of the resources consumed by the product. Simpson (1989) and Williamson (1992) compute bid prices using a linear programming approximation constructed under the assumption that the product requests take on their expected values. They use the optimal values of the dual variables associated with certain capacity availability constraints to measure the value of a unit of capacity. 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 scale linearly with the same rate. Talluri and van Ryzin (1999) use sampled realizations of the product requests in the linear program to capture some information about the distributions of the product requests. Bertsimas and Popescu (2003) measure the value of a unit of capacity directly using the change in the optimal objective value of the linear program in response to a change in the right side of a capacity availability constraint.

The value of a unit of capacity of a 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) uses linear value function approximations. Cooper and de Mello (2007) decompose the dynamic programming formulation of the problem by pairs of resources. Amaruchkul et al. (2007) develop bid price policies for cargo revenue management. Topaloglu (2009) computes upper bounds on the optimal total expected revenue, where he allows accepting a product request partially by consuming capacities only on some of the resources used by the product. Kunnumkal and Topaloglu (2010a) develop an approximation strategy by relaxing the capacity constraints through Lagrange multipliers. Zhang (2011) decomposes the dynamic programming formulation by the resources. Kirshner and Nediak (2015) use a second order code programming approximation. The approaches discussed in this paragraph yield either time or 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, where customers choose among the offered products. Gallego et al. (2004) give an analogue of the linear programming approximation under customer choice behavior. The number of decision variables in the linear program increases
exponentially with the number of products, so it is common to solve the linear program using column generation. Zhang and Cooper (2005, 2009) give bounds on the value functions when customers choose among the flight legs between the same origin and destination. Liu and van Ryzin (2008) develop approximations by decomposing the dynamic programming formulation of the problem by the resources. Kunnumkal and Topaloglu (2008) give a more refined linear programming approximation to capture the customer arrival process more accurately. Zhang and Adelman (2009) construct linear value function approximations. Bront et al. (2009) analyze the complexity of the column generation subproblem for the linear programming approximation when customers choose under a mixture of multinomial logit models. Mendez-Diaz et al. (2010) give valid cuts for the same subproblem. Kunnumkal and Topaloglu (2010b) heuristically decompose the problem by the resources by allocating the revenue from a product over the resources it uses. Meissner and Strauss (2012) construct separable and piecewise-linear approximations to the value functions. Meissner et al. (2012) observe that if there are multiple customer segments choosing according to different choice models, then the linear programming approximation can be challenging to solve and they develop tractable relaxations. Talluri (2014) tighten a similar relaxation using sampled customer arrivals and valid cuts. Kunnumkal and Talluri (2016b) theoretically compare the upper bounds on the optimal total expected revenue provided by the different methods in the existing literature. Strauss and Talluri (2017) give properties for the sets of products considered by different customer segments that ensure that the linear programming approximation can be solved tractably. Lastly, van Ryzin and Vulcano (2008a,b) use stochastic approximation to compute booking limits, and Topaloglu (2008), and Chaneton and Vulcano (2011) extend this work to computing bid prices. None of the work reviewed thus far provides constant-factor performance guarantees.

A number of papers characterize the loss in the optimal total expected revenue for the policies derived from linear programming approximations. All of this work considers an asymptotic regime, where the total expected demands and the capacities of the resources increase linearly with the same rate $k$. Cooper (2002) and Maglaras and Meissner (2006) bound the loss by $O(\sqrt{k})$, whereas Jasin and Kumar (2012) bound the loss by a constant independent of $k$, as long as we periodically solve the linear program over the selling horizon. Jasin and Kumar (2013) show that the standard bid price policy has a loss of at least $\Omega(\sqrt{k})$. We emphasize that these losses hold in an asymptotic regime as $k$ increases and they involve constants that are dependent on the input data, including the numbers of resources and products. Moreover, the losses are additive. In contrast, our performance guarantee of $1/(1 + L)$ is multiplicative, it holds without an asymptotic regime, and it does not depend on any input data other than $L$.

Lastly, the recent work by Wang et al. (2016), Gallego et al. (2016), and Rusmevichientong et al. (2017) develops policies with constant-factor performance guarantees for dynamic resource
allocation problems. In all of this work, however, each product request, if accepted, uses only one resource. In contrast, we consider here network revenue management problems where each product request consumes a combination of resources, which are significantly more challenging and rather nontrivial. The network revenue management setting requires building value function approximations that consider the interactions between the resources, and designing a recursion to update the coefficients of the basis functions in the value function approximations.

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 to customer choice behavior and products consuming multiple units 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

We have $m$ resources indexed by $L = \{1, \ldots, m\}$ and $n$ products indexed by $J = \{1, \ldots, n\}$. 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 $A_j \subseteq L$. We use $L$ to denote the maximum number of resources that can be used by a product, so $L = \max_{j \in J} |A_j|$. Accepting a request for product $j$ generates a revenue of $r_j$. We have $T$ time periods in the selling horizon indexed by $T = \{1, \ldots, T\}$. 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 $\lambda_{t,j}$. With the remaining probability $1 - \sum_{j \in J} \lambda_{t,j}$, there is no request for a product.

Our goal is to find a policy to determine which product request 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 $\mathbf{x} = (x_1, \ldots, x_m) \in \mathbb{Z}_+^m$ to capture the state of the resources. The set of possible states is $Q = \{ \mathbf{x} \in \mathbb{Z}_+^m : x_i \leq C_i \ \forall \ i \in L \}$. We can accept a request for product $j$ if 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 $\prod_{i \in A_j} \mathbb{1}_{x_i \geq 1} = 1$.

We use $V_t(\mathbf{x})$ to denote the maximum total expected revenue over time periods $t, t+1, \ldots, T$, given that the system is in state $\mathbf{x}$ at time period $t$. Letting $e_i \in \{0, 1\}^m$ denote the $i^{th}$ unit vector
and defining $[a]^+ = \max\{a, 0\}$, we can find the optimal policy by computing the optimal value functions $\{V^t(\mathbf{x}) : \mathbf{x} \in \mathcal{Q}, \ t \in \mathcal{T}\}$ through the dynamic program

$$V^t(\mathbf{x}) = \sum_{j \in \mathcal{J}} \lambda_j^t \left( \prod_{i \in \mathcal{A}_j} 1_{x_i \geq 1} \right) \max \left\{ r_j + V^{t+1} \left( \mathbf{x} - \sum_{i \in \mathcal{A}_j} \mathbf{e}_i \right), V^{t+1}(\mathbf{x}) \right\}$$

$$+ \left( 1 - \sum_{j \in \mathcal{J}} \lambda_j^t + \sum_{j \in \mathcal{J}} \lambda_j^t \left( 1 - \prod_{i \in \mathcal{A}_j} 1_{x_i \geq 1} \right) \right) V^{t+1}(\mathbf{x})$$

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

(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 $\mathbf{x} - \sum_{i \in \mathcal{A}_j} \mathbf{e}_i$. If we reject, then we do not generate revenue and the state of the resources at the next time period remains at $\mathbf{x}$. In addition, 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 $\mathbf{x}$. The second equality above follows simply by arranging the terms. By the dynamic program above, given that the state of the resources at time period $t$ is $\mathbf{x}$, if $r_j \geq V^{t+1}(\mathbf{x}) - V^{t+1}(\mathbf{x} - \sum_{i \in \mathcal{A}_j} \mathbf{e}_i)$, then it is optimal to accept a request for product $j$ as long as we have capacity on all resources used by product $j$. Letting $\mathbf{C} = (C_1, \ldots, C_m)$ be the initial resource capacities, the optimal total expected revenue is $V^1(\mathbf{C})$.

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. Our approximate policy is guided by these value function approximations. 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,
where, for each subset $A \subseteq \mathcal{L}$ of resources that can be used by a product, we have a basis function $\varphi_A : \mathcal{Q} \to [0, 1]$. In particular, the value function approximation $H^t$ is given by

$$H^t(x) = \sum_{j \in \mathcal{J}} \gamma^t_j \varphi_{A_j}(x), \tag{2}$$

where $\mathcal{B} = \{\varphi_A : A \subseteq \mathcal{L}\}$ is a prespecified collection of basis functions indexed by subsets of resources and $\{\gamma^t_j : j \in \mathcal{J}, t \in \mathcal{T}\}$ are adjustable coefficients. If no product uses the subset of resources $A$, then $\varphi_A$ is not needed in (2). We refer to our basis functions as availability-tracking basis functions because we will impose the condition that $\varphi_A(x)$ takes the value of zero if the vector of resource capacities $x$ does not provide enough availability to serve a request for a product using the subset of resources $A$. In the next definition, we fully specify the conditions that we impose on availability-tracking basis functions.

**Definition 3.1 (Availability-Tracking Bases)** The collection $\mathcal{B} = \{\varphi_A : A \subseteq \mathcal{L}\}$ is called a collection of availability-tracking basis functions if it satisfies the following conditions.

(a) **Availability Tracking:** For each subset $A \subseteq \mathcal{L}$ and $x \in \mathcal{Q}$, $\varphi_A(x) = 0$ whenever $x_i = 0$ for some $i \in A$. That is, $\varphi_A(x) \leq \prod_{i \in A} 1_{\{x_i \geq 1\}}$. 

(b) **Limited Dependence:** For each subset $A \subseteq \mathcal{L}$ and $x \in \mathcal{Q}$, the value of the basis function $\varphi_A(x)$ only depends on $(x_i : i \in A)$. That is, $\varphi_A(x) = \varphi_A(y)$ whenever $x_i = y_i$ for all $i \in A$.

(c) **Normalization:** For each subset $A \subseteq \mathcal{L}$, we have $\varphi_A(C) = 1$.

As discussed right before the definition, the range of $\varphi_A$ is the interval $[0, 1]$, and the first property ensures that $\varphi_A(x)$ takes the value of zero if the resource capacities $x$ are not sufficient to serve a request for a product using the subset of resources $A$. The second property ensures that $\varphi_A(x)$ is independent of the components of the resource capacity vector $x$ that are not in $A$. The third property ensures that $\varphi_A(x)$ is one when the resource capacities $x$ are at their largest possible values. For example, the minimum basis function $\varphi_A(x) = \min_{i \in A} \frac{x_i}{C_i}$ and the polynomial basis function $\varphi_A(x) = \prod_{i \in A} \frac{x_i}{C_i}$ satisfy the three properties in the definition above.

Given a collection $\mathcal{B} = \{\varphi_A : A \subseteq \mathcal{L}\}$ of availability-tracking basis functions, we define the maximum scaled incremental contribution $\Delta_\mathcal{B}$ from a unit of resource as

$$\Delta_\mathcal{B} = \max_{i \in \mathcal{L}, A \subseteq \mathcal{L}} \max_{x \in \mathcal{Q} : x_i \geq 1} C_i \times (\varphi_A(x) - \varphi_A(x - e_i)).$$

A collection with a smaller value of $\Delta_\mathcal{B}$ will result in an approximate policy with a better performance guarantee, but $\Delta_\mathcal{B}$ can never be smaller than one, as shown in the next lemma.
Lemma 3.2 If $\mathcal{B} = \{\varphi_A : A \subseteq \mathcal{L}\}$ is a collection of availability-tracking basis functions, then we must have $\Delta_{\mathcal{B}} \geq 1$.

Proof: Consider an arbitrary subset $A \subseteq \mathcal{L}$ and a resource $i \in A$. By parts (a) and (c) of Definition 3.1, we have $\varphi_A(C - C_i e_i) = 0$ and $\varphi_A(C) = 1$, so by a telescoping sum, we get

$$1 = \varphi_A(C) - \varphi_A(C - C_i e_i) = \sum_{h=1}^{C_i} \varphi_A \left( \sum_{s \neq i} C_s e_s + h e_i \right) - \varphi_A \left( \sum_{s \neq i} C_s e_s + (h-1) e_i \right) \leq \sum_{h=1}^{C_i} \frac{\Delta_{\mathcal{B}}}{C_i} = \Delta_{\mathcal{B}},$$

where the inequality follows from the definition of $\Delta_{\mathcal{B}}$. 

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

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

$$\varphi_A(x) - \varphi_A(x - e_i) = \min_{\ell \in A} \left\{ \frac{x_{\ell}}{C_{\ell}} \right\} - \min_{\ell \in A} \left\{ \frac{x_{\ell} - \mathbb{I}_{\ell=i}}{C_{\ell}} \right\} = \begin{cases} \frac{1}{C_i} \min_{\ell \in A} \frac{x_{\ell}}{C_{\ell}} & \text{if } \frac{x_i}{C_i} = \min_{\ell \in A} \frac{x_{\ell}}{C_{\ell}} \\ \frac{x_i - 1}{C_i} \min_{\ell \in A} \frac{x_{\ell}}{C_{\ell}} & \text{if } \frac{x_i}{C_i} > \frac{x_i - 1}{C_i} \min_{\ell \in A} \frac{x_{\ell}}{C_{\ell}} \\ 0 & \text{if } \frac{x_i - 1}{C_i} \min_{\ell \in A} \frac{x_{\ell}}{C_{\ell}} \end{cases}$$

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

Example 3.4 (Polynomial) Let $\varphi_A(x) = \prod_{i \in A} \frac{x_i}{C_i}$. It is simple to verify that $\mathcal{B} = \{\varphi_A : A \subseteq \mathcal{L}\}$ satisfies the three properties in Definition 3.1, so $\Delta_{\mathcal{B}} \geq 1$ by Lemma 3.2. Furthermore,

$$\varphi_A(x) - \varphi_A(x - e_i) = \left( \prod_{\ell \in A} \frac{x_{\ell}}{C_{\ell}} \right) - \left( \prod_{\ell \in A \setminus \{i\}} \frac{x_{\ell} - 1}{C_{\ell}} \right) \frac{x_i}{C_i} \leq \frac{1}{C_i} \prod_{\ell \in A \setminus \{i\}} \frac{x_{\ell}}{C_{\ell}} \leq \frac{1}{C_i},$$

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

In the next lemma, we show that we can construct availability-tracking basis functions from existing ones by taking a composition with a continuously differentiable function.

Lemma 3.5 (Constructing Bases) Let $\mathcal{B} = \{\varphi_A : A \subseteq \mathcal{L}\}$ be a collection of availability-tracking basis functions such that $\varphi_A$ is componentwise nondecreasing for all $A \subseteq \mathcal{L}$. If $f : [0, 1] \to [0, 1]$ is a differentiable function with $f(0) = 0$, $f(1) = 1$ and $\max_{a \in [0, 1]} f'(a) < \infty$, then $\mathcal{B}' = \{f \circ \varphi_A : A \subseteq \mathcal{L}\}$ is also a collection of availability-tracking basis functions with $\Delta_{\mathcal{B}'} \leq (\max_{a \in [0, 1]} f'(a)) \times \Delta_{\mathcal{B}}$. 
Proof: It is simple to verify that $\mathcal{B}'$ satisfies the three properties in Definition 3.1. Moreover, by the mean value theorem, for each $x \in Q$ with $x_i \geq 1$, there exists $y \in [0,1]$ such that
\[
f(x) = f(y) \leq \left( \max_{a \in [0,1]} f'(a) \right) \times (x - e_i),
\]
where the last inequality follows from our assumption that $\varphi$ is componentwise nondecreasing, so $\varphi(x) - \varphi(x - e_i) \geq 0$. The inequality above implies that $\Delta_{\mathcal{B}'} \leq \left( \max_{a \in [0,1]} f'(a) \right) \times \Delta_{\mathcal{B}}$. \hfill \qed

Since $f(0) = 0$ and $f(1) = 1$, we must have $\max_{a \in [0,1]} f'(a) \geq 1$ in the lemma. So, the lemma does not imply that $\Delta_{\mathcal{B}'} \leq \Delta_{\mathcal{B}}$. One can check that if $\varphi(x)$ is a linear function of $x$, then it does not satisfy the properties in Definition 3.1. Thus, linear basis functions are not availability-tracking.

### 3.2 Approximate Policy and Performance Guarantee

In this section, we give an algorithm to compute the coefficients $\{\gamma^t_j : j \in J, \ t \in T\}$ in the value function approximations in (2). This algorithm requires a recursive computation over the time periods in the selling horizon. Once we construct our value function approximations, the greedy policy with respect to the value function approximations yields our approximate policy. Then, we give a performance guarantee for our approximate policy. To compute the coefficients $\{\gamma^t_j : j \in J, \ t \in T\}$ in the value function approximations in (2), we use the algorithm below.

- **Initialization**: Let $\mathcal{B} = \{\varphi_A : A \subseteq L\}$ be any collection of availability-tracking basis functions and $\theta \geq 0$ be a tuning parameter. Initialize $\gamma^{T+1}_j = 0$ for all $j \in J$.

- **Coefficient Computation**: For each $t = T, T-1, \ldots, 1$, use the coefficients $\{\gamma^{t+1}_j : j \in J\}$ to compute $\{\gamma^t_j : j \in J\}$ as
\[
\gamma^t_j = \gamma^{t+1}_j + \lambda^t_j \left[ r_j - \theta \sum_{i \in A_j} \frac{1}{C_i} \sum_{k \in J} \mathbb{1}_{\{i \in A^k\}} \gamma^{t+1}_k \right].
\]

The above algorithm allows us to compute $\{\gamma^t_j : j \in J, \ t \in T\}$, which specifies the approximate value functions $\{H^t : t \in T\}$ given in (2).

Given that the state of the resources at time period $t$ is $x$, if $r_j \geq V^{t+1}(x) - V^{t+1}(x - \sum_{i \in A_j} e_i)$, then it is optimal to accept a request for product $j$ as long as we have capacity on all the resources used by product $j$. In other words, if $r_j \geq V^{t+1}(x) - V^{t+1}(x - \sum_{i \in A_j} e_i)$, then it is optimal to accept a request for product $j$ as long as we have $\prod_{i \in A_j} \mathbb{1}_{\{x_i \geq 1\}} = 1$. We obtain our approximate policy by replacing $V^{t+1}$ in the last inequality with $H^{t+1}$, which corresponds to the greedy policy with respect to the value function approximations $\{H^t : t \in T\}$.

To formally state our approximate policy, we use $u^{\text{App},t}(x) = \{u^{\text{App},t}_j(x) : j \in J\}$ to denote the decision function of the approximate policy at time period $t$. Given that the state of the
resources at time period $t$ is $x$, we have $u_{j,t}^{\text{App}}(x) = 1$ if we accept a request for product $j$ at time period $t$. Otherwise, we have $u_{j,t}^{\text{App}}(x) = 0$. Thus, $u_{j,t}^{\text{App}}(x)$ is given by

$$u_{j,t}^{\text{App}}(x) = \begin{cases} \prod_{i \in A_j} 1_{\{x_i \geq 1\}} & \text{if } r_j \geq H_{t+1}(x) - H_{t+1} \left( x - \sum_{i \in A_j} e_i \right), \\ 0 & \text{otherwise.} \end{cases}$$ (4)

The next theorem gives a performance guarantee for our approximate policy as a function of the tuning parameter $\theta$ in (3) and the maximum number of resources $L$ used by a product.

**Theorem 3.6 (Performance)** If the tuning parameter $\theta$ satisfies $\theta \geq \Delta_B$, then the total expected revenue obtained by the approximate policy is at least $1/(1 + \theta L)$ of the optimal.

We give the proof of Theorem 3.6 in the next section. To obtain the best performance guarantee, we need to choose the tuning parameter $\theta$ as small as possible and the smallest possible value of $\theta$ in the theorem is $\Delta_B$. By Lemma 3.2, we have $\Delta_B \geq 1$, but as shown in Examples 3.3 and 3.4, there are choices of basis functions under which $\Delta_B = 1$. Therefore, working with these basis functions, we can choose the tuning parameter $\theta$ as one and obtain an approximate policy whose total expected revenue is at least $1/(1 + L)$ of the optimal. 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, Theorem 3.6 provides a constant-factor approximation guarantee. Moreover, although we obtain the best performance guarantee by choosing $\theta$ at its smallest possible value of $\Delta_B$, our computational experiments indicate that increasing $\theta$ beyond $\Delta_B$ can improve the total expected revenue of the approximate policy. We view the tuning parameter $\theta$ as a knob that provides flexibility in the implementation of our approximate policy. Lastly, as is the case for most constant-factor approximation algorithms, the performance guarantee in Theorem 3.6 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.

We provide some insight into the computation of the coefficients $\{\gamma_{j,t} : j \in J, \ t \in T\}$. Letting $\mathcal{L} = \{1, \ldots, m\}$, we consider the case with $m + 1$ resources indexed by $\{0\} \cup \mathcal{L}$ and $m$ products indexed by $\mathcal{L}$. The capacity of resource 0 is $C_0$ and the capacities of all other resources are one. Product $j$ uses resources 0 and $j$. Thus, resource 0 is a common resource used by all products. For each $j \in \mathcal{L}$, resource $j$ has one unit of capacity, so we can serve at most one request
for product \( j \), but it may be optimal to reject a request for product \( j \) because serving such a request also consumes the common resource. For this problem instance, (3) reduces to

\[
\gamma_j^t = \gamma_j^{t+1} + \lambda_j^t \left[ r_j - \frac{1}{C_0} \sum_{k \in \mathcal{L}} \gamma_k^{t+1} \gamma_j^{t+1} \right] = \lambda_j^t \max \left\{ r_j - \frac{1}{C_0} \sum_{k \in \mathcal{L}} \gamma_k^{t+1} \gamma_j^{t+1} \right\} + (1 - \lambda_j^t) \gamma_j^{t+1},
\]

where we choose the tuning parameter \( \theta \) as one. Intuitively, \( \gamma_j^t \) captures the net expected profit contribution from product \( j \) at time period \( t \). With probability \( \lambda_j^t \), we have a request for product \( j \) at time period \( t \). If we accept the request, then we generate a revenue of \( r_j \), but we also consume a unit of the common resource, which limits our ability to serve a request for another product in the future. The opportunity cost associated with the consumption of the common resource appears as \( \frac{1}{C_0} \sum_{k \in \mathcal{L}} \gamma_k^{t+1} \) in our algorithm, yielding the term \( r_j - \frac{1}{C_0} \sum_{k \in \mathcal{L}} \gamma_k^{t+1} \) in the max operator. As \( C_0 \) increases, \( \frac{1}{C_0} \sum_{k \in \mathcal{L}} \gamma_k^{t+1} \) decreases, reflecting the intuition that if we have more of the common resource, then accepting the request less severely limits our ability to serve a request for another product in the future. If we have a request for product \( j \) at time period \( t \), then we have the option of rejecting this request, in which case, the net expected profit contribution from product \( j \) at time period \( t \) is the same as the net expected profit contribution at the next time period, yielding the term \( \gamma_j^{t+1} \) in the max operator. Similarly, with probability \( 1 - \lambda_j^t \), we do not get a request for product \( j \) at time period \( t \), yielding the term \( (1 - \lambda_j^t) \gamma_j^{t+1} \) on the right side.

### 3.3 Proof of Theorem 3.6

The proof of Theorem 3.6 makes use of two lemmas. The next lemma bounds the opportunity cost of accepting a request for product \( j \) under our approximate value functions.

**Lemma 3.7 (Bound on Opportunity Cost)** For a collection of availability-tracking basis functions \( \mathcal{B} = \{ \varphi_A : A \subseteq \mathcal{L} \} \), let \( H(\mathbf{x}) = \sum_{k \in \mathcal{J}} \gamma_k \varphi_{Ak}(\mathbf{x}) \), where the coefficients \( \{ \gamma_k : k \in \mathcal{J} \} \) satisfy \( \gamma_k \geq 0 \) for all \( k \in \mathcal{J} \). Then, for each \( j \in \mathcal{J} \) and \( \mathbf{x} \in \mathcal{Q} \) such that \( \mathbf{x} - \sum_{i \in A^j} \mathbf{e}_i \geq 0 \), we have

\[
H(\mathbf{x}) - H(\mathbf{x} - \sum_{i \in A^j} \mathbf{e}_i) \leq \Delta_{\varphi} \sum_{i \in A^j} \frac{1}{C_i} \sum_{k \in \mathcal{J}} 1_{\{i \in A^k\}} \gamma_k.
\]

**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

\[
H(\mathbf{x}) - H(\mathbf{x} - \mathbf{e}_i) = \sum_{k \in \mathcal{J}} 1_{\{i \in A^k\}} \gamma_k (\varphi_{Ak}(\mathbf{x}) - \varphi_{Ak}(\mathbf{x} - \mathbf{e}_i)) \leq \frac{\Delta_{\varphi}}{C_i} \sum_{k \in \mathcal{J}} 1_{\{i \in A^k\}} \gamma_k,
\]

where the equality holds because \( \varphi_{Ak}(\mathbf{x}) - \varphi_{Ak}(\mathbf{x} - \mathbf{e}_i) = 0 \) whenever \( i \notin A^k \) by the second property in Definition 3.1 and the inequality follows from the definition of \( \Delta_{\varphi} \). Thus, the base case holds.
Suppose that the result holds for any $|A^t| \leq s$. Consider a case in which $|A^t| = s+1$, so $A^t = B \cup \{\ell\}$ for some $B \subseteq L$ with $|B| = s$ and $\ell \in L$ with $\ell \notin B$. Letting $y = x - \sum_{i \in B} e_i$, we obtain

$$H(x) - H\left( x - \sum_{i \in A^t} e_i \right) = H(x) - H\left( x - \sum_{i \in B} e_i - e_\ell \right)$$

$$= H(x) - H\left( x - \sum_{i \in B} e_i \right) + H\left( x - \sum_{i \in B} e_i \right) - H\left( x - \sum_{i \in B} e_i - e_\ell \right)$$

$$\leq \Delta_{\mathcal{B}} \sum_{i \in B} \frac{1}{C_i} \sum_{k \in \mathcal{J}} 1_{i \in A^k} \gamma_k + H(y) - H(y - e_\ell)$$

$$\leq \Delta_{\mathcal{B}} \sum_{i \in B} \frac{1}{C_i} \sum_{k \in \mathcal{J}} 1_{i \in A^k} \gamma_k + \frac{\Delta_{\mathcal{B}}}{C_\ell} \sum_{k \in \mathcal{J}} 1_{(\ell \in A^k)} \gamma_k$$

$$= \Delta_{\mathcal{B}} \sum_{i \in A^t} \frac{1}{C_i} \sum_{k \in \mathcal{J}} 1_{i \in A^k} \gamma_k,$$

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 $A^t = B \cup \{\ell\}$. □

Thus, $\theta \sum_{i \in A^t} \frac{1}{C_i} \sum_{k \in \mathcal{J}} 1_{i \in A^k} \gamma_k$ in (3) is an upper bound on $H(x) - H(x - \sum_{i \in A^t} e_i)$ for any $\theta \geq \Delta_{\mathcal{B}}$. In the next lemma, we bound the optimal total expected revenue using $\{H^t : t \in \mathcal{T}\}$.

**Lemma 3.8 (Upper Bound on Optimal Total Expected Revenue)** If the value function approximations $\{H^t : t \in \mathcal{T}\}$ are constructed using (3), then $V^1(C) \leq (1 + \theta L) H^1(C)$.

**Proof:** Noting the dynamic program in (1), it is a well-known result that the optimal total expected revenue is given by the optimal objective value of the linear program

$$\min \ \tilde{V}^1(C)$$

s.t. $\tilde{V}^t(x) \geq \tilde{V}^{t+1}(x) + \sum_{j \in \mathcal{J}} \lambda^t_j \sum_{i \in A^t} 1_{\{i \geq 1\}} \left[ r_j - \tilde{V}^{t+1}(x) + \tilde{V}^{t+1}(x - \sum_{i \in A^t} e_i) \right] \forall x \in \mathcal{Q}, \ t \in \mathcal{T},$

where the decision variables are $\{\tilde{V}^t(x) : x \in \mathcal{Q}, \ t \in \mathcal{T}\}$; see Adelman (2007). In the linear program above, we follow the convention that $\tilde{V}^{T+1}(x) = 0$ for all $x \in \mathcal{Q}$. Since we minimize the objective function in the linear program above, if $\{\nu^t(x) : x \in \mathcal{Q}, \ t \in \mathcal{T}\}$ is a feasible solution to the linear program above, then $\nu^1(C)$ is an upper bound on the optimal total expected revenue. Letting $\{\alpha^t : t \in \mathcal{T}\}$ and $\{\beta^t_i : i \in \mathcal{L}, \ t \in \mathcal{T}\}$ be defined as

$$\alpha^t = \sum_{j \in \mathcal{J}} \gamma^t_j \quad \text{and} \quad \beta^t_i = \frac{\theta}{C_i} \sum_{k \in \mathcal{J}} 1_{i \in A^k} \gamma_k,$$

we construct the solution $\{\nu^t(x) : x \in \mathcal{Q}, \ t \in \mathcal{T}\}$ to the linear program as $\nu^t(x) = \alpha^t + \sum_{i \in \mathcal{L}} \beta^t_i x_i$. We will now show that $\{\nu^t(x) : x \in \mathcal{Q}, \ t \in \mathcal{T}\}$ is a feasible solution to the linear program above. It
follows from (3) that \( \gamma_j^t \geq \gamma_j^{t+1} \), so we have \( \beta_i^t \geq \beta_i^{t+1} \). Moreover, noting that \( \nu^{t+1}(x) \) is linear in \( x \), we have \( \nu^{t+1}(x) - \nu^{t+1}(x - \sum_{i \in A^t} e_i) = \sum_{i \in A^t} \beta_i^{t+1} \). In this case, evaluating the right side of the constraint in the linear program at the solution \( \{\nu^t(x) : x \in Q, t \in T\} \), we get

\[
\nu^{t+1}(x) + \sum_{j \in J} \lambda_j^t \left( \prod_{i \in A^t} 1_{\{x_i \geq 1\}} \right) \left[ r_j - \nu^{t+1}(x) + \nu^{t+1}(x - \sum_{i \in A^t} e_i) \right]^+ \\
= \sum_{j \in J} \gamma_j^{t+1} + \sum_{i \in L} \beta_i^{t+1} x_i + \sum_{j \in J} \lambda_j^t \left( \prod_{i \in A^t} 1_{\{x_i \geq 1\}} \right) \left[ r_j - \sum_{i \in A^t} \beta_i^{t+1} \right]^+ \\
= \sum_{j \in J} \gamma_j^{t+1} + \sum_{i \in L} \beta_i^{t+1} x_i + \sum_{j \in J} \lambda_j^t \left( \prod_{i \in A^t} 1_{\{x_i \geq 1\}} \right) \left[ r_j - \theta \sum_{i \in A^t} C_i \sum_{k \in J} 1_{\{i \in A^k\}} \gamma_k^{t+1} \right]^+ \\
= \sum_{j \in J} \gamma_j^{t+1} + \sum_{i \in L} \beta_i^{t+1} x_i + \sum_{j \in J} \lambda_j^t \left( \prod_{i \in A^t} 1_{\{x_i \geq 1\}} \right) (\gamma_j^t - \gamma_j^{t+1}) \\
\leq \sum_{j \in J} \gamma_j^t + \sum_{i \in L} \beta_i^{t+1} x_i = \nu^t(x),
\]

where the second equality uses the definition of \( \beta_i^{t+1} \), the third equality follows from (3), and the inequality uses the fact that \( \beta_i^{t+1} \leq \beta_i^t \) along with the observation that \( \gamma_j^t - \gamma_j^{t+1} \geq 0 \), so \( (\prod_{i \in A^t} 1_{\{x_i \leq 1\}}) (\gamma_j^t - \gamma_j^{t+1}) \leq \gamma_j^t - \gamma_j^{t+1} \). The chain of inequalities above shows that the solution \( \{\nu^t(x) : x \in Q, t \in T\} \) is feasible for the linear program. Thus, \( \nu^t(C) \) is an upper bound on the optimal total expected revenue, satisfying \( \nu^t(C) \geq V^1(C) \). In this case, we have

\[
V^1(C) \leq \nu^t(C) = \alpha^1 + \sum_{i \in L} \beta_i^1 C_i \\
= \sum_{j \in J} \gamma_j^1 + \theta \sum_{i \in L} \sum_{k \in J} 1_{\{i \in A^k\}} \gamma_k^1 \\
= \sum_{j \in J} \gamma_j^1 + \theta \sum_{k \in J} \sum_{i \in L} 1_{\{i \in A^k\}} \\
= \sum_{j \in J} \gamma_j^1 + \theta \sum_{k \in J} |A^k| \\
\leq (1 + \theta L) \sum_{j \in J} \gamma_j^1 = (1 + \theta L) H^1(C),
\]

where the last inequality follows because \( |A^j| \leq L \) for all \( j \in J \) and the last equality holds since we have \( \varphi_A(C) = 1 \) for all \( A \subseteq L \) by the third property in Definition 3.1, in which case, it follows that \( \sum_{j \in J} \gamma_j^1 = \sum_{j \in J} \gamma_j^1 \varphi_A(C) = H^1(C) \).

To compute the total expected revenue obtained by the approximate policy, we can use a dynamic programming recursion similar to (1). Let \( U^t(x) \) be the total expected revenue obtained by the approximate policy over time periods \( t, t+1, \ldots, T \) given that the system is in state \( x \) at time
period \( t \). Noting the decision function for the approximate policy in (4), we can compute \( \{ U^t : t \in T \} \) through the recursion

\[
U^t(x) = \sum_{j \in J} \lambda_j^t u_j^{\text{App},t}(x) \left[ r_j + U^{t+1} \left( x - \sum_{i \in A^j} e_i \right) \right] + \left( 1 - \sum_{j \in J} \lambda_j^t + \sum_{j \in J} \lambda_j^t (1 - u_j^{\text{App},t}(x)) \right) U^{t+1}(x)
\]

\[
= U^{t+1}(x) + \sum_{j \in J} \lambda_j^t u_j^{\text{App},t}(x) \left[ r_j - U^{t+1}(x) + U^{t+1} \left( x - \sum_{i \in A^j} e_i \right) \right],
\]

(5)

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 \( \lambda_j^t \), in which case, if we have \( u_j^{\text{App},t}(x) = 1 \) 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 \( x - \sum_{i \in A^j} e_i \). If there is a request for product \( j \) at time period \( t \), but we have \( u_j^{\text{App},t}(x) = 0 \), then the state of the resources at the next time period remains at \( x \). Similarly, with probability \( 1 - \sum_{j \in J} \lambda_j^t \), there is no request for a product at time period \( t \), in which case, the state of the resources at the next time period remains at \( x \) as well. The second equality above follows simply by arranging the terms.

Using \( \{ U^t : t \in T \} \) computed as above, the total expected revenue obtained by the approximate policy is given by \( U^1(C) \). Observe that the coefficients of \( U^{t+1} \) on the right side of the first equality in (5) are all nonnegative. Therefore, if we replace \( U^{t+1} \) on the right side of the first equality in (5) with a function \( G^{t+1} \) satisfying \( G^{t+1} \leq U^{t+1} \), then the expression on right side of the first equality becomes smaller. In this case, since the expression on the right side of the second equality in (5) is obtained by arranging the terms in expression on the right side of the first equality, it follows that if we replace \( U^{t+1} \) on the right side of the second equality in (5) with a function \( G^{t+1} \) satisfying \( G^{t+1} \leq U^{t+1} \), then the expression on the right side of the second equality in (5) becomes smaller as well. This observation will become useful when we give the proof of Theorem 3.6 below.

**Proof of Theorem 3.6:** In the proof, we will show that the inequality \( U^t(x) \geq H^t(x) \) holds for all \( x \in Q \) and \( t \in T \), where \( U^t(x) \) is given by (5) and \( H^t(x) \) is the value function approximation in (2) with the coefficients \( \{ \gamma_j^t : j \in J \} \) computed through the recursion in (3). Then, applying this inequality at the first time period with the initial capacities of all of the resources, it follows that \( U^1(C) \geq H^1(C) \geq V^1(C)/(1 + \theta L) \), where the second inequality follows from Lemma 3.8. Thus, the total expected revenue obtained by the approximate policy is at least \( 1/(1 + \theta L) \) of the optimal total expected revenue, which is the desired result.

We now use induction over the time periods in the selling horizon to prove that \( U^t(x) \geq H^t(x) \) holds for all \( x \in Q \) and \( t \in T \). Consider the base case at time period \( T + 1 \). Since \( U^{T+1} = H^{T+1} = 0 \), the base case holds. Suppose that the results hold at time period \( t + 1 \), so \( U^{t+1}(x) \geq H^{t+1}(x) \) for
all \( x \in Q \). Since \( H^{t+1} \leq U^{t+1} \), replacing \( U^{t+1} \) on the right side of the second inequality in (5) with \( H^{t+1} \) and noting the discussion right before the proof, we have

\[
U^t(x) \geq H^{t+1}(x) + \sum_{j \in J} \lambda_j^t u^\text{App}(x) \left[ r_j - H^{t+1}(x) + H^{t+1}(x - \sum_{i \in A_j} e_i) \right] \\
= H^{t+1}(x) + \sum_{j \in J} \lambda_j^t \left( \prod_{i \in A_j} 1_{\{i \geq 1\}} \right) \left[ r_j - H^{t+1}(x) + H^{t+1}(x - \sum_{i \in A_j} e_i) \right] \\
\geq H^{t+1}(x) + \sum_{j \in J} \lambda_j^t \left( \prod_{i \in A_j} 1_{\{i \geq 1\}} \right) \left[ r_j - \Delta \sum_{i \in A_j} \frac{1}{C_i} \sum_{k \in J} 1_{\{i \in A_k\}} \gamma_{k}^{t+1} \right] \\
\geq H^{t+1}(x) + \sum_{j \in J} \varphi_{A_j}(x) (\gamma_j^t - \gamma_j^{t+1}) \\
\geq H^{t+1}(x) + \sum_{j \in J} \varphi_{A_j}(x) (\gamma_j^t - \gamma_j^{t+1}) = H^t(x),
\]

where the first equality above follows because the definition of the decision function of the approximate policy in (4) implies that \( u^\text{App}(x) = 1 \) if and only if \( \prod_{i \in A_j} 1_{\{i \geq 1\}} = 1 \) and \( r_j - H^{t+1}(x) + H^{t+1}(x - \sum_{i \in A_j} e_i) \geq 0 \). The second inequality follows from Lemma 3.7. The third inequality is due to the fact that the tuning parameter \( \theta \) is chosen such that \( \theta \geq \Delta \). The second equality follows from the definition of \( \gamma_j^t \) in (3). The fourth inequality follows from the first property in Definition 3.1, along with the fact that \( \gamma_j^t - \gamma_j^{t+1} \geq 0 \). The last equality holds because we have \( H^{t+1}(x) = \sum_{j \in J} \gamma_j^{t+1} \varphi_{A_j}(x) \) by the definition of the value function approximations. By the chain of inequalities above, we have \( U^t(x) \geq H^t(x) \), completing the induction argument. \( \square \)

### 3.4 Tightness of the Analysis

We describe a problem instance to demonstrate that the performance guarantee in Theorem 3.6 is tight. We consider a problem instance with a single resource. All products use this resource. Thus, the maximum number of resources \( L \) used by a product is one. Noting Examples 3.3 and 3.4, if we use the minimum or polynomial basis functions, then we have \( \Delta = 1 \), in which case, by Theorem 3.6, we can choose the tuning parameter \( \theta \) as one. With \( L = 1 \) and \( \theta = 1 \), Theorem 3.6 implies that the total expected revenue obtained by our approximate policy is at least \( 1/2 \) of the optimal. We give a problem instance where the ratio between the total expected revenue from our approximate policy and the optimal total expected revenue is arbitrarily close to \( 1/2 \).

There are \( T \) time periods in the selling horizon. We have a single resource. The capacity of the resource is given by \( C_1 = T \). We have two products indexed by \( J = \{1, 2\} \). The revenues associated
with the two products are \( r_1 = \frac{1}{T} (1 - \frac{1}{T}) \) and \( r_2 = 1 \). At the first \( T - 1 \) time periods, we have a request for product 1 at each time period with probability one. At the last time period \( T \), we have a request for product 2 with probability one. That is, we have

\[
\lambda_i^t = \begin{cases} 
1 & \text{if } t < T \\
0 & \text{if } t = T 
\end{cases}
\]

and

\[
\lambda_2^t = \begin{cases} 
0 & \text{if } t < T \\
1 & \text{if } t = T 
\end{cases}
\]

Since the capacity of the resource is equal to the number of time periods, we never run out of capacity by accepting the requests. Thus, the optimal policy accepts all requests, in which case, the total expected revenue of the optimal policy is \( \text{OPT} = (T - 1) r_1 + r_2 = \frac{(T - 1)^2}{T^2} + 1 \).

Considering our approximate policy, since we have a single resource and both products use this resource, letting \( x = (x_1) \) denote the remaining capacity of the resource, we have

\[
\min_{i \in A^j} \frac{x_i}{c_i} = \prod_{i \in A^j} \frac{x_i}{c_i} = \frac{x_1}{c_1} = \frac{x}{T}
\]

for all \( j = 1, 2 \). Therefore, irrespective of whether we use the minimum or polynomial basis functions, the value function approximation in (2) is given by

\[
H' (x) = \sum_{j \in \{1, 2\}} \gamma_j^t \frac{x_j}{T},
\]

which implies that

\[
H'(x) - H'(x - e_1) = \frac{1}{T} (\gamma_1^t + \gamma_2^t).
\]

In this case, using the fact that we never run out of capacity by accepting the requests for the products, the decision function of our approximate policy given in (4) can be written as \( u_j^{\text{app},t}(x) = 1 \) if and only if \( r_j \geq \frac{1}{T} (\gamma_1^{t+1} + \gamma_2^{t+1}) \).

Observe that the decision function of our approximate policy in this problem instance depends on the sum \( \gamma_1^{t+1} + \gamma_2^{t+1} \), but not on the individual values of \( \gamma_1^{t+1} \) and \( \gamma_2^{t+1} \). Next, we compute the total expected revenue obtained by our approximate policy.

Since there is a single resource with capacity \( C_1 = T \) and we choose \( \theta = 1 \), the recursion in (3) takes the form

\[
\gamma_j^t = \gamma_j^{t+1} + \lambda_j^t \left[ r_j - \frac{1}{T} \sum_{k \in \{1, 2\}} \gamma_k^{t+1} \right] +
\]

for all \( j = 1, 2 \) and \( t \in \mathcal{T} \). In this case, noting that the decision function of our approximate policy depends on the sum \( \gamma_1^t + \gamma_2^t \), adding the last recursion over all \( j = 1, 2 \) and letting \( \Gamma^t = \gamma_1^t + \gamma_2^t \), we write this recursion as

\[
\Gamma^t = \Gamma^{t+1} + \sum_{j \in \{1, 2\}} \lambda_j^t \left[ r_j - \frac{1}{T} \Gamma^{t+1} \right]^+,
\]

with the boundary condition that \( \Gamma^{T+1} = 0 \). Since \( \lambda_2^T = 1 \) and \( r_2 = 1 \), from the recursion above, we get \( \Gamma^T = 1 \). Since \( \lambda_1^{T-1} = 1, r_1 = \frac{1}{T} (1 - \frac{1}{T}) \) and \( \Gamma^T = 1 \), we get \( \Gamma^{T-1} = 1 + \left[ \frac{1}{T} (1 - \frac{1}{T}) - \frac{1}{T} \right]^+ = 1 \). Since \( \lambda_1^{T-2} = 1, r_1 = \frac{1}{T} (1 - \frac{1}{T}) \) and \( \Gamma^{T-1} = 1 \), we get \( \Gamma^{T-2} = 1 + \left[ \frac{1}{T} (1 - \frac{1}{T}) - \frac{1}{T} \right]^+ = 1 \). Continuing in a similar fashion, it follows that \( \Gamma^t = 1 \) for all \( t \in \mathcal{T} \).

At each one of the time periods \( t = 1, \ldots, T - 1 \), we have a request for product 1 with probability one. Since \( r_1 = \frac{1}{T} (1 - \frac{1}{T}) < \frac{1}{T} \), \( \Gamma^{T+1} = \frac{1}{T} (\gamma_1^{T+1} + \gamma_2^{T+1}) \), our approximate policy rejects the requests for product 1 at the time periods \( t = 1, \ldots, T - 1 \). At time period \( T \), we have a request for product 2 with probability one. Since \( r_2 = 1 \geq 0 = \frac{1}{T} \), \( \Gamma^{T+1} = \frac{1}{T} (\gamma_1^{T+1} + \gamma_2^{T+1}) \), our approximate policy accepts the request for product 2 at time period \( T \). Thus, the total expected revenue from
our approximate policy is $APP = r_2 = 1$. In this case, the ratio between the total expected revenue from our approximate policy and the optimal total expected revenue is $\frac{APP}{OPT} = \frac{1}{(\frac{T-1}{T^2}) + 1}$, which becomes arbitrarily close to $1/2$ as $T$ becomes arbitrarily large.

4. Extensions

We extend our approximate policy to 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 programming approximation 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. Here, 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 for the model in Section 2. We only discuss the additional notation that we need. If we offer the subset $S \subseteq J$ of products to a customer arriving at time period $t$, then the customer chooses product $j \in S$ with probability $\phi^f_t(S)$. Naturally, we have $\phi^f_t(S) = 0$ for all $j \notin S$. Note that the choices of the customers at different time periods may be governed by different purchase probabilities. We refer to a subset of products that we offer to the customers as an assortment. We use $\mathcal{F} \subseteq 2^J$ 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.

**Assumption 4.1 (Substitutability and Feasibility)** *For all $t \in T$, $S \in \mathcal{F}$, $j \in S$, and $k \notin S$, we have $\phi^f_t(S) \geq \phi^f_t(S \cup \{k\})$. Moreover, 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 $V^t(x)$ 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 $x$. We can compute the optimal value functions $\{V^t(x) : x \in Q, \ t \in T\}$ using the dynamic program

\[
V^t(x) = \max_{S \in F} \left\{ \sum_{j \in J} \phi_j^t(S) \left( \prod_{i \in A_j} 1_{\{x_i \geq 1\}} \right) \left[ r_j + V^{t+1} \left( x - \sum_{i \in A_j} e_i \right) \right] + \left( 1 - \sum_{j \in J} \phi_j^t(S) + \sum_{j \in J} \phi_j^t(S) \left( 1 - \prod_{i \in A_j} 1_{\{x_i \geq 1\}} \right) \right) V^{t+1}(x) \right\}
\]

\[
= V^{t+1}(x) + \max_{S \in F} \left\{ \sum_{j \in J} \phi_j^t(S) \left( \prod_{i \in A_j} 1_{\{x_i \geq 1\}} \right) \left( r_j - V^{t+1}(x) + V^{t+1} \left( x - \sum_{i \in A_j} e_i \right) \right) \right\}, \tag{6}
\]

with the boundary condition that $V^{T+1} = 0$. In the dynamic program above, there is a customer arrival at each time period with probability one. If there is a strictly positive probability of no customer arrival at time period $t$, then letting $\lambda^t \in [0, 1)$ be the customer arrival probability at time period $t$, all we need to do is to replace the choice probability $\phi_j^t(S)$ in the dynamic program above with the choice probability $\tilde{\phi}_j^t(S) = \lambda^t \phi_j^t(S)$.

In the first equality in (6), 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 $\prod_{i \in A_j} 1_{\{x_i \geq 1\}} = 1$, 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 $x - \sum_{i \in A_j} e_i$. On the other hand, the arriving customer does not choose any product and decides to leave without a purchase with probability $1 - \sum_{j \in J} \phi_j^t(S)$, in which case, the state of the resources at the next time period remains at $x$. 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 $\prod_{i \in A_j} 1_{\{x_i \geq 1\}} = 0$, then the customer leaves without a purchase, in which case, the state of the resources at the next time period remains at $x$ as well. In (6), 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 (6), 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 $r_j - V^{t+1}(x) + V^{t+1}(x - \sum_{i \in A_j} e_i) < 0$, in which case, by Assumption 4.1, the choice probabilities of all other remaining products in the assortment do not decrease, yielding another assortment that provides an objective value that is as large as the original one.

As noted previously, computing the optimal value functions $\{V^t : t \in T\}$ is intractable. We use a value function approximation of the form $H^t(x) = \sum_{j \in J} \gamma_j^t \varphi_j(x)$ where $B = \{\varphi_A : A \subseteq L\}$ is a
collection of availability-tracking basis functions. We compute the coefficients \( \{\gamma_t^j : j \in J, t \in T\} \) in the value function approximations using a slight variation of our earlier algorithm.

**Initialization:** Let \( \mathcal{B} = \{\varphi_A : A \subseteq L\} \) be any collection of availability-tracking basis functions and \( \theta \geq 0 \) be a tuning parameter. Initialize \( \gamma_{T+1}^j = 0 \) for all \( j \in J \).

**Coefficient Computation:** For each \( t = T, T-1, \ldots, 1 \), use the coefficients \( \{\gamma_t^j : j \in J\} \) to compute the assortment \( \hat{S}_t \in F \) at time period \( t \) as

\[
\hat{S}_t = \arg \max_{S \in F} \left\{ \sum_{j \in J} \phi_j^t(S) \left( r_j - \theta \sum_{i \in A_j} \frac{1}{C_i} \sum_{k \in J} \mathbf{1}_{\{i \in A_k\}} \gamma_k^{t+1} \right) \right\}.
\]

(7)

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

\[
\gamma_j^t = \gamma_j^{t+1} + \phi_j^t(\hat{S}_t) \left( r_j - \theta \sum_{i \in A_j} \frac{1}{C_i} \sum_{k \in J} \mathbf{1}_{\{i \in A_k\}} \gamma_k^{t+1} \right).
\]

(8)

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

Given that the state of the resources at time period \( t \) is \( x \), we solve the maximization problem on the right side of the second equality in (6) to find the optimal assortment to offer. We construct our approximate policy by replacing \( V_t^{t+1} \) in this problem with \( H_t^{t+1} \). In this case, given that the state of the resources at time period \( t \) is \( x \), our approximate policy offers the assortment

\[
S^{\text{App},t}(x) = \arg \max_{S \in F} \left\{ \sum_{j \in J} \phi_j^t(S) \left( \prod_{i \in A_j} \mathbf{1}_{\{x_i \geq 1\}} \right) \left( r_j - H_t^{t+1}(x) + H_t^{t+1}(x - \sum_{i \in A_j} e_i) \right) \right\}.
\]

(9)

An optimal solution to the problem above can be viewed as the decision function of our approximate policy under customer choice behavior. Applying the following theorem, our approximate policy enjoys the same performance guarantee as in Theorem 3.6. The proof of this theorem uses a technique similar to the one in Section 3.3. We defer the proof to Appendix A.

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

### 4.2 Multiple Units of Capacity Consumption

In the model that we described in Section 2, each product consumes at most one unit of each resource. However, we can extend our approach to allow products to use multiple units of each
resource. Once again, the notation that we use closely follows the one introduced for the model 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 nonnegative 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 from consideration. To find a policy that maximizes the total expected revenue over the selling horizon, 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 \( \prod_{i \in \mathcal{A}^j} \mathbb{1}_{\{z_i \geq 1\}} \) with \( \prod_{i \in \mathcal{A}^j} \mathbb{1}_{\{z_i \geq a_{ij}\}} \) and all occurrences of \( \sum_{i \in \mathcal{A}^j} e_i \) with \( \sum_{i \in \mathcal{A}^j} a_{ij} e_i \) in the dynamic program.

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

\[
H^t(x) = \sum_{j \in \mathcal{J}} \gamma^t_j \varphi_{A^j}(\mathcal{G}_j(x)) .
\] (10)

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

- **Initialization:** Let \( \mathcal{B} = \{ \varphi_A : A \subseteq \mathcal{L} \} \) be any collection of availability-tracking basis functions and \( \theta \geq 0 \) be a tuning parameter. Initialize \( \gamma^T_{j+1} = 0 \) for all \( j \in \mathcal{J} \).

- **Coefficient Computation:** For each \( t = T, T - 1, \ldots, 1 \), use the coefficients \( \{ \gamma^{t+1}_j : j \in \mathcal{J} \} \) to compute \( \{ \gamma^t_j : j \in \mathcal{J} \} \) as

\[
\gamma^t_j = \gamma^{t+1}_j + \lambda_j \left[ r_j - \theta \sum_{i \in \mathcal{A}^j} \frac{2m_i - 1}{C_i} \sum_{k \in \mathcal{J}} \mathbb{1}_{\{i \in \mathcal{A}^k\}} \gamma^{t+1}_k \right] .
\] (11)

To construct our approximate policy, we use the decision function in (4) after replacing \( \prod_{i \in \mathcal{A}^j} \mathbb{1}_{\{z_i \geq 1\}} \) with \( \prod_{i \in \mathcal{A}^j} \mathbb{1}_{\{z_i \geq a_{ij}\}} \) and \( \sum_{i \in \mathcal{A}^j} e_i \) with \( \sum_{i \in \mathcal{A}^j} a_{ij} e_i \). This decision function provides the decisions made by our approximate policy given that the state of the resources at time period \( t \) is \( x \). 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 B.
Theorem 4.3 (Performance under Multiple Units Consumption) Let $M = \max_{i \in L, j \in J} a_{ij}$ be the maximum number of units of a resource used by a product. If the tuning parameter $\theta$ satisfies $\theta \geq \Delta_B$, then the total expected revenue obtained by the approximate policy is at least $1/(1 + \theta(2M - 1)L)$ of the optimal.

The approach that we use to compute the coefficients $\{\gamma_t^j : j \in J, t \in T\}$ in (11) and the performance guarantee in Theorem 4.3 reduce to the approach in (3) and the performance guarantee in Theorem 3.6 when each product uses at most one unit of a resource because, in the latter case, we have $m_i = 1$ for all $i \in L$, $M = 1$, and $G_j(x) = x$ for all $x \in Q$.

4.3 Leveraging a Linear Programming Approximation

It is common to formulate linear programming approximations to network revenue management problems under the assumption that the arrivals of the customers take on their expected values. The optimal objective values of such linear programming approximations can provide upper bounds on the optimal total expected revenues, which become useful when assessing the performance of various heuristics. We can leverage an optimal solution to a linear programming approximation to come up with an approximate policy with the same performance guarantee as in Theorem 3.6. We explain the idea using the model given 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 the model given in Section 2, the optimal total expected revenue is $V^1(C)$, where the value functions $\{V_t : t \in T\}$ are obtained through the dynamic program in (1). To formulate the linear program, we use the decision variables $\{z_j : j \in J\}$, where $z_j$ is the expected number of requests that we accept for product $j$. Using $\Lambda_j = \sum_{t \in T} \lambda_j^t$ to denote the total expected number of requests for product $j$ over the whole selling horizon, we consider the linear program

$$\max \left\{ \sum_{j \in J} r_j z_j : \sum_{j \in J} 1_{\{i \in A_j\}} z_j \leq C_i \ \forall \ i \in L, \ 0 \leq z_j \leq \Lambda_j \ \forall \ j \in J \right\}. \tag{12}$$

The objective function above accounts for the total expected revenue over the selling horizon. The first constraint ensures that the total expected capacity consumptions of the resources do not exceed the initial capacities. The second constraint ensures that the expected numbers of requests that we serve for the products do not exceed the expected demands.

It is well known that the optimal objective value of the linear program in (12) provides an upper bound on the optimal total expected revenue $V^1(C)$; see Bertsimas and Popescu (2003). For a collection of availability-tracking basis functions $B = \{\varphi_A : A \subseteq L\}$, we approximate the optimal value functions $\{V^t : t \in T\}$ using value function approximations $\{H^t : t \in T\}$ of
the form $H^t(x) = \sum_{j \in \mathcal{J}} \gamma^t_j \varphi_{A_j}(x)$. We use the following algorithm to compute the coefficients \( \{\gamma^t_j : j \in \mathcal{J}, t \in T\} \) in the value function approximations.

**Initialization:** Let $\mathcal{B} = \{\varphi_A : A \subseteq L\}$ be any collection of availability-tracking basis functions, \( \theta \geq 0 \) be a tuning parameter, and \( \{z^*_j : j \in \mathcal{J}\} \) be an optimal solution to the linear program in (12). Initialize \( \gamma^{T+1}_j = 0 \) for all \( j \in \mathcal{J} \).

**Coefficient Computation:** For each \( t = T, T-1, \ldots, 1 \), use the coefficients \( \{\gamma^{t+1}_j : j \in \mathcal{J}\} \) to compute \( \{\gamma^t_j : j \in \mathcal{J}\} \) as

\[
\gamma^t_j = \gamma^{t+1}_j + \frac{z^*_j}{\lambda_j} \left[ r_j - \theta \sum_{i \in A_j} \frac{1}{C_i} \sum_{k \in \mathcal{J}} 1_{\{i \in A_k\}} \gamma^{t+1}_k \right].
\]

Once we construct the value function approximations \( \{H^t : t \in T\} \) using the algorithm above, we use the same decision function in (4) in our approximate policy. The following theorem gives a performance guarantee for our approximate policy. The proof is in Appendix C.

**Theorem 4.4 (Performance with Linear Programming Approximation)** If the tuning parameter \( \theta \) satisfies \( \theta \geq \Delta_B \), then the total expected revenue obtained by the approximate policy is at least \( 1/(1 + \theta L) \) 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 (13), the coefficients \( \{\gamma^t_j : t \in T\} \) for this product do not contribute significantly to the value function approximation.

## 5. Computational Experiments

In this section, we describe computational experiments we conducted on a collection of test problems to assess the numerical performance of our approximate policy.

### 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), adopted 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 itineraries in the airline network. In our test problems, we have a single hub and \( N \) 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. 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. For each origin-destination pair, there are two itineraries, high-fare and low-fare. Therefore, 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 $\kappa$ times the one associated with the corresponding low-fare itinerary. We vary $\kappa$ in our computational experiments as well. Recalling that $\lambda_{tj}$ is the probability that we have a request for product $j$ at time period $t$, if product $j$ corresponds to a high-fare itinerary, then $\lambda_{tj}$ is an increasing function of $t$, whereas if product $j$ corresponds to a low-fare itinerary, then $\lambda_{tj}$ is a decreasing function of $t$. Therefore, the requests for the high-fare itineraries tend to arrive later in the selling horizon, which makes it important to reserve the capacities for the high-fare itinerary requests by rejecting the requests for the low-fare itineraries early in the selling horizon. Noting that $\sum_{i \in T} \sum_{j \in J} 1_{\{i \in A_j\}} \lambda_{tj}$ gives the total expected demand for the capacity on flight leg $i$, the initial capacity of flight leg $i$ is set to be $C_i = \frac{1}{\alpha} \sum_{i \in T} \sum_{j \in J} 1_{\{i \in A_j\}} \lambda_{tj}$. Thus, larger values for $\alpha$ yield tighter capacities. We vary $\alpha$ in our computational experiments.

Letting $N$, $\kappa$, and $\alpha$ be as above, and recalling that $T$ is the length of the selling horizon, we vary $T \in \{200, 600\}$, $N \in \{4, 5, 6, 8\}$, $\kappa \in \{2, 3\}$, and $\alpha \in \{1.0, 1.2, 1.6\}$, to get 48 test problems.

### 5.2 Benchmark Methods

We compare two benchmark methods. The first one is our approximate policy. The second one is based on the linear programming approximation in (12).

**Approximate Policy:** This benchmark corresponds to our approximate policy with the decision function given in (4). We refer to this benchmark as AP, standing for approximate policy. In all our computational experiments, we use the minimum basis functions given in Example 3.3. We repeated our computational experiments with polynomial basis functions given in Example 3.4 as well, but the performance of AP under the polynomial basis functions was slightly inferior. In our practical implementation of AP, 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 $\bm{x}$, 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 $\{\gamma_{jt} : j \in J, t = (k - 1) \frac{T}{5} + 1, \ldots, T\}$. 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 $\theta$ at the beginning of each segment. The values of the coefficients $\{\gamma^j_t : j \in J, \ t \in \mathbb{T}\}$ in (3) depend on $\theta$, which, in turn, implies that the total expected revenue obtained by AP also depends on $\theta$. When reconstructing our value function approximations at the beginning of each segment, we search for the best tuning parameter over the interval $[1,15]$ with a precision of 0.01. Given that we use the tuning parameter $\theta$ when reconstructing our value function approximations at the beginning of segment $k$, let $U^{k,\theta}(x)$ be the total expected revenue obtained by AP over time periods $(k-1)\frac{T}{N}+1,\ldots,T$ starting with the capacities $x$ for the flight legs. Computing the total expected revenue $U^{\theta,t}(x)$ exactly is intractable, but we estimate this quantity using simulation. At the beginning of segment $k$, we choose the value of the tuning parameter $\theta$ as $\arg \max \{U^{k,\theta}(x) : \theta \in \{1,1.01,1.02,\ldots,15\}\}$. We use this value for the tuning parameter until we reach the beginning of the next segment.

The performance guarantee for AP that we give in Theorem 3.6 holds as long as $\theta \geq \Delta_{\mathbb{B}}$ and it improves when we use smaller values for the tuning parameter $\theta$. According to Example 3.3, we have $\Delta_{\mathbb{B}} = 1$ for the minimum basis functions. So, we can choose the tuning parameter $\theta$ as one to obtain the best possible performance guarantee for AP, but our computational experiments indicated that although setting $\theta = 1$ gives the best possible theoretical performance guarantee, choosing a different value for the tuning parameter may provide better practical performance.

**Bid Price Policy:** This benchmark is the standard bid price policy in the network revenue management literature; see Section 3.3 in Talluri and van Ryzin (2005). We refer to this benchmark as BP, standing for bid price policy. The main idea behind BP is to use the optimal values of the dual variables associated with the first constraint in the linear program in (12) 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 $\{\mu^*_i : i \in \mathbb{L}\}$ be the optimal values of the dual variables associated with the
first constraint in problem (12), we use $\mu^*_i$ to capture the value of a unit of capacity on flight leg $i$. Therefore, BP 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 BP, we make two modifications. First, we divide the selling horizon into five equal segments and resolve problem (12) 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 $x$, then we replace $C_i$ with $x_i$ and $\Lambda_j$ with $\sum_{t=(k-1)T+1}^T \lambda^*_t$ in problem (12). Letting $\{\mu^*_i : i \in L\}$ be the optimal values of the dual variables associated with the first constraint, we use these values of the dual variables when making the decisions over segment $k$. Second, to treat AP and BP fairly, we introduce a tuning parameter in BP as well. Using $\zeta$ to denote the tuning parameter, BP accepts a request for itinerary $j$ if and only if $r_j \geq \zeta \sum_{i \in A_j} \mu^*_i$ and there are sufficient capacities to serve a request for itinerary $j$. Following the same approach for AP, we calibrate the value of the tuning parameter $\zeta$ at the beginning of each segment, where we use simulation to estimate the total expected revenue obtained by BP over the remaining portion of the selling horizon with a certain choice of $\zeta$. We exhaustively search for the best tuning parameter over the interval $[0.5, 1.5]$ with a precision of 0.01. We choose BP as the benchmark because it is one of the most commonly used methods in academia and industry. Our goal is to demonstrate that AP is competitive with such a commonly used benchmark.

5.3 Computational Results

Table 1 shows the computational results on test problems with $T = 200$ time periods in the selling horizon, while Table 2 shows the computational results on 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 $(T, N, \alpha, \kappa)$, where $N$, $\alpha$ and $\kappa$ are as given in the discussion of 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 (12). In the third column, we show the total expected revenue obtained by AP. In the fourth column, we show the total expected revenue obtained by BP. We estimate these total expected revenues by simulating the performance of each benchmark over 100 sample paths. In the fifth column, we give the percent gap between the total expected revenues obtained by AP and BP. In the sixth column, we give a 95% confidence interval for the gap between the total expected revenues obtained by AP and BP after normalizing the total expected revenues by the upper bound on the optimal total expected revenue. In other words, using $\{\text{APR}^*_s : s = 1, \ldots, 100\}$ and $\{\text{BPR}^*_s : s = 1, \ldots, 100\}$ to denote the total revenues obtained by AP and BP for a particular problem instance in the 100 sample paths, and letting $\text{UB}$ denote the upper bound on the optimal total expected revenue, we build a 95% confidence interval for the gap
between the performance of AP and BP by using the data \(\{100 \times \frac{\text{APR}^s - \text{BPR}^s}{\text{UB}} : s = 1, \ldots, 100\}\). If the confidence interval is included in the interval \([0, \infty)\), then the gap between the two benchmarks is statistically significant at 95% level in favor of AP. If the confidence interval includes zero, then the gap between the two benchmarks is not statistically significant.

In 45 out of 48 test problems, the total expected revenue obtained by AP is larger than that obtained by BP and the gap is statistically significant. In the remaining three test problems, there is no statistically significant gap. Over all the test problems, the average percent gap between the total expected revenues obtained by AP and BP is 1.78%. To underline a few trends, as the parameter \(\alpha\) increases and the capacities on the flight legs get tighter, the performance gap between AP and BP increases. 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 AP and BP are, respectively, 1.34%, 1.67%, and 2.31%. 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 AP does a better job of capturing this tradeoff. As the parameter \(\kappa\) increases and the revenue difference between the high-fare and low-fare itineraries increases, the performance gap between AP and BP 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

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>((200, 4, 1.0, 4.0))</td>
<td>21,531</td>
<td>19,848</td>
<td>0.27</td>
<td>([-0.22, 0.77])</td>
</tr>
<tr>
<td>((200, 4, 1.0, 8.0))</td>
<td>34,571</td>
<td>32,344</td>
<td>0.05</td>
<td>([0.13, 1.18])</td>
</tr>
<tr>
<td>((200, 4, 1.2, 4.0))</td>
<td>19,882</td>
<td>18,306</td>
<td>1.13</td>
<td>([0.58, 1.69])</td>
</tr>
<tr>
<td>((200, 4, 1.2, 8.0))</td>
<td>32,922</td>
<td>30,791</td>
<td>1.75</td>
<td>([1.04, 2.46])</td>
</tr>
<tr>
<td>((200, 4, 1.6, 4.0))</td>
<td>17,530</td>
<td>15,963</td>
<td>1.46</td>
<td>([0.83, 2.08])</td>
</tr>
<tr>
<td>((200, 4, 1.6, 8.0))</td>
<td>30,570</td>
<td>28,523</td>
<td>2.64</td>
<td>([1.88, 3.41])</td>
</tr>
<tr>
<td>((200, 5, 1.0, 4.0))</td>
<td>22,144</td>
<td>20,904</td>
<td>0.70</td>
<td>([0.24, 1.16])</td>
</tr>
<tr>
<td>((200, 5, 1.0, 8.0))</td>
<td>35,387</td>
<td>33,829</td>
<td>1.78</td>
<td>([1.13, 2.43])</td>
</tr>
<tr>
<td>((200, 5, 1.2, 4.0))</td>
<td>21,263</td>
<td>19,491</td>
<td>0.80</td>
<td>([0.20, 1.40])</td>
</tr>
<tr>
<td>((200, 5, 1.2, 8.0))</td>
<td>34,495</td>
<td>32,208</td>
<td>2.35</td>
<td>([1.63, 3.07])</td>
</tr>
<tr>
<td>((200, 5, 1.6, 4.0))</td>
<td>18,870</td>
<td>16,811</td>
<td>1.03</td>
<td>([0.37, 1.69])</td>
</tr>
<tr>
<td>((200, 5, 1.6, 8.0))</td>
<td>32,081</td>
<td>29,353</td>
<td>2.44</td>
<td>([1.66, 3.23])</td>
</tr>
<tr>
<td>((200, 6, 1.0, 4.0))</td>
<td>22,300</td>
<td>20,448</td>
<td>0.72</td>
<td>([0.20, 1.24])</td>
</tr>
<tr>
<td>((200, 6, 1.0, 8.0))</td>
<td>35,544</td>
<td>33,124</td>
<td>1.73</td>
<td>([0.97, 2.49])</td>
</tr>
<tr>
<td>((200, 6, 1.2, 4.0))</td>
<td>20,932</td>
<td>18,819</td>
<td>0.73</td>
<td>([0.13, 1.32])</td>
</tr>
<tr>
<td>((200, 6, 1.2, 8.0))</td>
<td>34,172</td>
<td>31,425</td>
<td>2.64</td>
<td>([1.81, 3.47])</td>
</tr>
<tr>
<td>((200, 6, 1.6, 4.0))</td>
<td>18,592</td>
<td>16,460</td>
<td>1.98</td>
<td>([1.25, 2.72])</td>
</tr>
<tr>
<td>((200, 6, 1.6, 8.0))</td>
<td>31,824</td>
<td>28,928</td>
<td>3.01</td>
<td>([2.15, 3.88])</td>
</tr>
<tr>
<td>((200, 8, 1.0, 4.0))</td>
<td>20,052</td>
<td>18,207</td>
<td>1.07</td>
<td>([0.56, 1.58])</td>
</tr>
<tr>
<td>((200, 8, 1.0, 8.0))</td>
<td>31,835</td>
<td>29,613</td>
<td>3.19</td>
<td>([2.57, 3.82])</td>
</tr>
<tr>
<td>((200, 8, 1.2, 4.0))</td>
<td>18,952</td>
<td>16,739</td>
<td>1.34</td>
<td>([0.76, 1.92])</td>
</tr>
<tr>
<td>((200, 8, 1.2, 8.0))</td>
<td>30,727</td>
<td>28,015</td>
<td>3.50</td>
<td>([2.82, 4.18])</td>
</tr>
<tr>
<td>((200, 8, 1.6, 4.0))</td>
<td>16,833</td>
<td>14,492</td>
<td>1.83</td>
<td>([1.14, 2.53])</td>
</tr>
<tr>
<td>((200, 8, 1.6, 8.0))</td>
<td>28,608</td>
<td>25,734</td>
<td>5.01</td>
<td>([4.19, 5.83])</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td></td>
<td></td>
<td></td>
<td><strong>1.82</strong></td>
</tr>
</tbody>
</table>

Table 1: Computational results for the test problems with \(T = 200\) time periods in the selling horizon.
We proposed a constant-factor approximation algorithm for network revenue management problems when the number of resources used by a product is uniformly bounded. To our knowledge, our approximate policy is the first constant-factor approximation algorithm for network revenue management problems.
management problems. Our performance guarantee is independent of the input data, with no hidden constants. There are several interesting research directions to pursue. In this paper, 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.

In our model, there is no overbooking. One heuristic method to handle overbooking in the existing literature is to inflate the physically available resource capacities by overbooking pads to obtain virtual capacities. In this case, one uses a model under the assumption that overbooking is not possible, but the capacities of the resources correspond to their virtual capacities. Naturally, our model applies heuristically when we replace the physically available resource capacities with virtual capacities, but we lose our performance guarantee. A more sound approach is to start with a dynamic programming formulation of the network revenue management problem under overbooking and develop an approximate policy with a performance guarantee. The dynamic programming formulation under overbooking is fundamentally different since the state variable needs to keep track of the number of accepted requests for each product. Thus, extending our work to handle overbooking is not straightforward. Moreover, our approximate policy is not a bid price policy, as it does not explicitly estimate the value of a unit of resource. One can investigate developing bid price policies with constant-factor performance guarantees. Bid prices are of interest by themselves, but they are also used to heuristically decompose dynamic programming formulations of network revenue management problems by the resources. Bid prices computed through an approach similar to ours could be used in such decomposition heuristics. Lastly, our choice of the basis functions was based on numerical 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.

References


This page is intentionally blank. Proper e-companion title page, with INFORMS branding and exact metadata of the main paper, will be produced by the INFORMS office when the issue is being assembled.
Online Appendix:

A Constant-Factor Approximation Algorithm for Network Revenue Management

Appendix A: Proof of Theorem 4.2

The following lemma shows that the coefficients \( \{ \gamma_j^t : j \in J, \ t \in T \} \) computed through (8) are increasing in \( t \). This lemma is a consequence of Assumption 4.1.

**Lemma A.1** For all \( j \in J \) and \( t \in T \), we have \( \gamma_j^t \geq \gamma_j^{t+1} \).

**Proof:** Considering problem (7), we define \( \mu_j = r_j - \theta \sum_{i \in A^j} \frac{1}{C_i} \sum_{k \in J} \mathbb{1}_{i \in A^k} \gamma_k^{t+1} \) for notational brevity. We claim that if \( \hat{S}^t \) is an optimal solution to problem (7), then we have \( \phi_j^t(\hat{S}^t) \mu_j \geq 0 \) for all \( j \in \hat{S}^t \). We prove this claim by contradiction. Suppose, on the contrary, that the set \( \hat{N}^t = \{ j \in \hat{S}^t : \phi_j^t(\hat{S}^t) \mu_j < 0 \} \) is nonempty. Letting \( \hat{P}^t = \{ j \in \hat{S}^t \setminus \hat{N}^t : \mu_j \geq 0 \} \), we have \( \hat{P}^t \subseteq \hat{S}^t \), in which case, by Assumption 4.1, we have \( \phi_j^t(\hat{P}^t) \geq \phi_j^t(\hat{S}^t) \) 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

\[
\sum_{j \in \hat{S}^t} \phi_j^t(\hat{S}^t) \mu_j = \sum_{j \in \hat{S}^t \setminus \hat{N}^t} \phi_j^t(\hat{S}^t) \mu_j + \sum_{j \in \hat{N}^t} \phi_j^t(\hat{S}^t) \mu_j < \sum_{j \in \hat{S}^t \setminus \hat{N}^t} \phi_j^t(\hat{S}^t) \mu_j \leq \sum_{j \in \hat{P}^t} \phi_j^t(\hat{S}^t) \mu_j \leq \sum_{j \in \hat{P}^t} \phi_j^t(\hat{P}^t) \mu_j,
\]

where the first inequality uses the fact that \( \phi_j^t(\hat{S}^t) \mu_j < 0 \) for all \( j \in \hat{N}^t \) and \( \hat{N}^t \neq \emptyset \), 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 \( \phi_j^t(\hat{P}^t) \geq \phi_j^t(\hat{S}^t) \) and \( \mu_j \geq 0 \) for all \( j \in \hat{P}^t \). Since \( \hat{P}^t \in \mathcal{F} \), \( \hat{P}^t \) is a feasible solution to problem (7), in which case, the chain of inequalities above contradicts the fact that \( \hat{S}^t \) is an optimal solution to problem (7), establishing the claim. In this case, if \( j \in \hat{S}^t \), then (8) implies that \( \gamma_j^t = \gamma_j^{t+1} + \phi_j^t(\hat{S}^t) \mu_j \geq \gamma_j^{t+1} \). If \( j \notin \hat{S}^t \), then we have \( \phi_j^t(\hat{S}^t) = 0 \), so \( \gamma_j^t = \gamma_j^{t+1} + \phi_j^t(\hat{S}^t) \mu_j = \gamma_j^{t+1} \). \( \square \)

The following lemma gives an upper bound on the optimal total expected revenue computed through the dynamic program in (6). Its proof is similar to the one for Lemma 3.8.

**Lemma A.2** If the value function approximations \( \{ H^t : t \in T \} \) are constructed using (8), then \( V^1(C) \leq (1 + \theta L) H^1(C) \).
Proof: Using the decision variables \( \{ \tilde{V}^t(x) : x \in Q, \; t \in T \} \), the optimal total expected revenue \( V^1(C) \) is given by the optimal objective value of the linear program

\[
\begin{align*}
\text{min} & \quad \tilde{V}^1(C) \\
\text{s.t.} & \quad \tilde{V}^t(x) \geq \tilde{V}^{t+1}(x) + \sum_{j \in J} \phi_j(S) \left( \prod_{i \in A_j} 1_{\{x_i \geq 1\}} \right) \left( r_j - \tilde{V}^{t+1}(x) + \tilde{V}^{t+1} \left( x - \sum_{i \in A_j} e_i \right) \right) \\
& \quad \text{for all } x \in Q, \; t \in T, \; S \in F,
\end{align*}
\]

where we follow the convention that \( \tilde{V}^{T+1}(x) = 0 \) for all \( x \in Q \). We define \( \{ \nu^t(x) : x \in Q, \; t \in T \} \) as \( \nu^t(x) = \alpha^t + \sum_{i \in L} \beta_i^t x_i \), where \( \{ \alpha^t : t \in T \} \) and \( \{ \beta_i^t : i \in L, \; t \in T \} \) are given by

\[
\alpha^t = \sum_{j \in J} \gamma_j^t \quad \text{and} \quad \beta_i^t = \frac{\theta}{C_i} \sum_{k \in J} 1_{\{i \in A^k\}} \gamma_k^t.
\]

If we can show that \( \{ \nu^t(x) : x \in Q, \; t \in T \} \) is a feasible solution to the linear program above, then \( \nu^1(C) \) is an upper bound on the optimal objective value of this linear program.

To show the feasibility of the solution \( \{ \nu^t(x) : x \in Q, \; t \in T \} \) to the linear program above, computing the right side of the constraint in the linear program at this solution, we get

\[
\begin{align*}
\nu^{t+1}(x) + \sum_{j \in J} \phi_j(S) \left( \prod_{i \in A_j} 1_{\{x_i \geq 1\}} \right) \left( r_j - \nu^{t+1}(x) + \nu^{t+1} \left( x - \sum_{i \in A_j} e_i \right) \right) \\
& = \sum_{j \in J} \gamma_j^{t+1} + \sum_{i \in L} \beta_i^{t+1} x_i + \sum_{j \in J} \phi_j(S) \left( \prod_{i \in A_j} 1_{\{x_i \geq 1\}} \right) \left( r_j - \sum_{i \in A_j} \beta_i^{t+1} \right) \\
& = \sum_{j \in J} \gamma_j^{t+1} + \sum_{i \in L} \beta_i^{t+1} x_i + \sum_{j \in J} \phi_j(S) \left( \prod_{i \in A_j} 1_{\{x_i \geq 1\}} \right) \left( r_j - \theta \sum_{i \in A_j} 1_{\{i \in A^k\}} \gamma_k^{t+1} \right) \\
& \leq \sum_{j \in J} \gamma_j^{t+1} + \sum_{i \in L} \beta_i^{t+1} x_i + \max_{R \in F} \left\{ \sum_{j \in J} \phi_j(R) \left( \prod_{i \in A_j} 1_{\{x_i \geq 1\}} \right) \left( r_j - \theta \sum_{i \in A_j} 1_{\{i \in A^k\}} \gamma_k^{t+1} \right) \right\} \\
& \leq \sum_{j \in J} \gamma_j^{t+1} + \sum_{i \in L} \beta_i^{t+1} x_i + \max_{R \in F} \left\{ \sum_{j \in J} \phi_j(R) \left( r_j - \theta \sum_{i \in A_j} 1_{\{i \in A^k\}} \gamma_k^{t+1} \right) \right\} \\
& = \sum_{j \in J} \gamma_j^{t+1} + \sum_{i \in L} \beta_i^{t+1} x_i + \sum_{j \in J} \phi_j(S^t) \left( r_j - \theta \sum_{i \in A_j} 1_{\{i \in A^k\}} \gamma_k^{t+1} \right) \\
& = \sum_{j \in J} \gamma_j^{t+1} + \sum_{i \in L} \beta_i^{t+1} x_i + \sum_{j \in J} (\gamma_j^t - \gamma_j^{t+1}) \\
& \leq \sum_{j \in J} \gamma_j^t + \sum_{i \in L} \beta_i^t x_i \\
& = \nu^t(x).
\end{align*}
\]

In the chain of inequalities above, the first equality holds because \( \nu^{t+1}(x) \) is a linear function of \( x \) of the form \( \sum_{j \in J} \gamma_j^{t+1} + \sum_{i \in L} \beta_i^{t+1} x_i \), so \( \nu^{t+1}(x) - \nu^{t+1}(x - \sum_{i \in A_j} e_i) = \sum_{i \in A_j} \beta_i^{t+1} x_i \). The
second equality holds by the definition of $\beta_j^{t+1}$. To see that the second inequality holds, let $R^*$ be an optimal solution to the maximization problem on the left side of the inequality. Letting $\mu_j = r_j - \theta \sum_{i \in A} \frac{1}{C_i} \sum_{k \in J} \mathbb{1}_{\{i \in A^k\}} \gamma_k^{t+1}$ and $P^* = \{ j \in R^*: \mu_j \geq 0 \}$, so $\phi_j^t(R^*) \leq \phi_j^t(P^*)$ for all $j \in P^*$. The optimal objective value of the maximization problem satisfies $\sum_{j \in R^*} \phi_j^t(R^*) \left( \prod_{i \in A^j} \mathbb{1}_{\{x_i \geq 1\}} \right) \mu_j \leq \sum_{j \in P^*} \phi_j^t(R^*) \left( \prod_{i \in A^j} \mathbb{1}_{\{x_i \geq 1\}} \right) \mu_j$.

The third equality above follows from the definition of $\tilde{S}^t$. The fourth equality follows from (8). The last inequality holds because $\gamma_j^t \geq \gamma_j^{t+1}$ by Lemma A.1, so $\beta_j^t \geq \beta_j^{t+1}$ by the definition of $\beta_j^t$.

By the chain of inequalities above, $\{\nu^t(x): x \in Q, t \in T\}$ is a feasible solution to the linear program. Thus, $\nu^1(C)$ is an upper bound on the optimal objective value of the linear program, which is, in turn, equal to the optimal total expected revenue $V^1(C)$. In this case, we get

$$V^1(C) \leq \nu^1(C) = \alpha^1 + \sum_{i \in L} \beta_i^1 C_i,$$

where the second equality follows from the definition of $\beta_i^t$ and the last inequality follows because $|A^j| \leq L$ for all $j \in J$. The last equality follows from the third property in Definition 3.1 because $\varphi_A(C) = 1$ for all $A \subseteq L$, so $\sum_{j \in J} \gamma_j^1 = \sum_{j \in J} \gamma_j^1 \varphi_A(C) = H^1(C)$.

Let $U^t(x)$ denote the total expected revenue obtained by the approximate policy over time periods $t, t+1, \ldots, T$, given that the state of the resources at time period $t$ is $x$. Building on the dynamic program in (6), we can compute $\{U^t: t \in T\}$ through the recursion

$$U^t(x) = \sum_{j \in J} \phi_j^t(S^{App,t}(x)) \left( \prod_{i \in A^j} \mathbb{1}_{\{x_i \geq 1\}} \right) \left[ r_j + U^{t+1} \left( x - \sum_{i \in A^j} e_i \right) \right]$$

$$+ \left( 1 - \sum_{j \in J} \phi_j^t(S^{App,t}(x)) \right) \left( \sum_{j \in J} \phi_j^t(S^{App,t}(x)) \left( 1 - \prod_{i \in A^j} \mathbb{1}_{\{x_i \geq 1\}} \right) \right) U^{t+1}(x)$$

$$= U^{t+1}(x) + \sum_{j \in J} \phi_j^t(S^{App,t}(x)) \left( \prod_{i \in A^j} \mathbb{1}_{\{x_i \geq 1\}} \right) \left[ r_j - U^{t+1}(x) + U^{t+1} \left( x - \sum_{i \in A^j} e_i \right) \right],$$

with the boundary condition that $U^{T+1} = 0$. Recall that if the state of the remaining resources at time period $t$ is $x$, then the approximate policy offers the assortment $S^{App,t}(x)$ given in (9). The recursion above is analogous to the dynamic program in (6). The only difference is that the offered
assortment is fixed by the decision of the approximate policy. The total expected revenue obtained by the approximate policy is given by $U^1(C)$.

Here is the proof of Theorem 4.2.

**Proof of Theorem 4.2:** We use induction over the time periods to show that $U^t(x) \geq H^t(x)$ for all $x \in Q$ and $t \in T$. Considering the base case at time period $T+1$. Since $U^{T+1} = H^{T+1} = 0$, the base case holds. Suppose that the result holds at time period $t+1$, so that $U^{t+1}(x) \geq H^{t+1}(x)$ for all $x \in Q$. We proceed to showing that $U^t(x) \geq H^t(x)$ for all $x \in Q$. Since $H^{t+1} \leq U^{t+1}$, if we replace $U^{t+1}$ on the right side of the second equality in (14) with $H^{t+1}$, then precisely the same line of reasoning discussed right before the proof of Theorem 3.6 in the main text implies that the right side of the second equality in (14) gets smaller. Thus, we have

$$
U^t(x) \geq H^{t+1}(x) + \sum_{j \in J} \phi_j(S^{\text{App},t}(x)) \left( \prod_{i \in A_j} 1_{\{x_i \geq 1\}} \right) \left[ r_j - H^{t+1}(x) + H^{t+1}(x - \sum_{i \in A_j} e_i) \right] 
= H^{t+1}(x) + \max_{S \in F} \left\{ \sum_{j \in J} \phi_j(S) \left( \prod_{i \in A_j} 1_{\{x_i \geq 1\}} \right) r_j - H^{t+1}(x) + H^{t+1}(x - \sum_{i \in A_j} e_i) \right\} 
\geq H^{t+1}(x) + \max_{S \in F} \left\{ \sum_{j \in J} \phi_j(S) \left( \prod_{i \in A_j} 1_{\{x_i \geq 1\}} \right) r_j - \theta \sum_{i \in A_j} \frac{1}{C_i} \sum_{k \in J} 1_{\{i \in A_k\}} \gamma_k \right\} 
\geq H^{t+1}(x) + \max_{S \in F} \left\{ \prod_{i \in A_j} 1_{\{x_i \geq 1\}} \left( \gamma_j^t - \gamma_j^{t+1} \right) \right\} 
= H^{t+1}(x) + \sum_{j \in J} \phi_j(S^t) \left( \prod_{i \in A_j} 1_{\{x_i \geq 1\}} \right) \left( \gamma_j^t - \gamma_j^{t+1} \right) 
\geq H^{t+1}(x) + \sum_{j \in J} \varphi_{A_j}(x) \left( \gamma_j^t - \gamma_j^{t+1} \right) 
= H^t(x),
$$

where the first equality follows from the definition of the decision function for the approximate policy given in (9). The second inequality follows from Lemma 3.7. The third inequality is by the fact that $S^t \in F$ 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 (8). The fourth inequality follows by noting the first property in Definition 3.1 and using the fact that $\gamma_j^t \geq \gamma_j^{t+1}$ by Lemma A.1. The last equality holds since $H^t(x) = \sum_{j \in J} \gamma_j^t \varphi_{A_j}(x)$. The chain of inequalities above completes the induction argument, so $U_t(x) \geq H^t(x)$ for all $x \in Q$, $t \in T$.

The desired result follows by noting that $(1 + \theta L) U^1(C) \geq (1 + \theta L) H^1(C) \geq V^1(C)$, where the last inequality is by Lemma A.2. □
Appendix B: Proof of Theorem 4.3

Recall that product $j$ uses $a_{ij}$ units of resource $i$. We still use $A^i = \{i \in L : a_{ij} \geq 1\}$ 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 $\prod_{i \in A^j} \mathbb{I}_{\{x_i \geq a_{ij}\}}$ with $\prod_{i \in A^j} \mathbb{I}_{\{x_i \geq a_{ij}\}}$ and all occurrences of $\sum_{i \in A^j} e_i$ with $\sum_{i \in A^j} a_{ij} e_i$ in the dynamic program. In particular, we can compute the optimal policy by computing the value functions $\{V^t : t \in T\}$ through the dynamic program

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

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

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

**Lemma B.1** For a collection of availability-tracking basis functions $\mathcal{B} = \{\varphi_A : A \subseteq \mathcal{L}\}$, let $H(x) = \sum_{k \in \mathcal{J}} \gamma_k \varphi_{A^k}(G_k(x))$, where the coefficients $\{\gamma_k : k \in \mathcal{J}\}$ satisfy $\gamma_k \geq 0$ for all $k \in \mathcal{J}$. Then, for each $j \in \mathcal{J}$ and $x \in \mathcal{Q}$ such that $x - \sum_{i \in A^j} a_{ij} e_i \geq 0$, we have

$$H(x) - H\left(x - \sum_{i \in A^j} a_{ij} e_i\right) \leq \Delta_{\mathcal{B}} \sum_{i \in A^j} \frac{2m_i - 1}{C_i} \sum_{k \in \mathcal{J}} \mathbb{I}_{\{i \in A^k\}} \gamma_k.$$

**Proof:** Let $y = G_k(x)$ and $\tilde{y} = G_k(x - a_{ij} e_i)$. By the definition of $G_k$, note that $y$ and $\tilde{y}$ differ only in the $i^{th}$ component. Moreover, by the definition of $G_k$, we have

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

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 $m_i$, $a_{ij} + a_{ik} - 1 \leq 2m_i - 1$. So, by a telescoping sum, we get

$$\varphi_{A^k}(y) - \varphi_{A^k}(\tilde{y}) = \sum_{h=1}^{y_i - \tilde{y}_i} \left( \varphi_{A^k}(\tilde{y} + h e_i) - \varphi_{A^k}(\tilde{y} + (h - 1) e_i) \right) \leq \Delta_{\mathcal{B}} \frac{2m_i - 1}{C_i}, \quad (15)$$

where the inequality follows from the definition of $\Delta_{\mathcal{B}}$. 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 $g_k(x)$ and $g_k(x-a_{ij}e_i)$ differ only in the $i$-th component, by the second property in Definition 3.1, we have $\varphi_{Ak}(g_k(x)) - \varphi_{Ak}(g_k(x-a_{ij}e_i)) = 0$ whenever $i \not\in A^k$. In this case, we obtain the chain of inequalities

$$
H(x) - H(x - a_{ij}e_i) = \sum_{k \in \mathcal{J}} \mathbb{1}_{\{x \in A^k\}} \gamma_k \left( \varphi_{Ak}(g_k(x)) - \varphi_{Ak}(g_k(x-a_{ij}e_i)) \right)
$$

$$
\leq \frac{\Delta_{\mathcal{B}}(2m_i - 1)}{C_i} \sum_{k \in \mathcal{J}} \mathbb{1}_{\{A^k\}} \gamma_k,
$$

where the inequality is by (15). So, the base case holds. Suppose that the results holds for any $|A^j| \leq s$. 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 $\ell \not\in B$. Letting $w = x - \sum_{i \in A^j} a_{ij}e_i$ for notational brevity, we have

$$
H(x) - H \left( x - \sum_{i \in A^j} a_{ij}e_i \right) = H(x) - H \left( x - \sum_{i \in B} a_{ij}e_i - a_{ij}e_\ell \right)
$$

$$
= H(x) - H \left( x - \sum_{i \in B} a_{ij}e_i \right) + H \left( x - \sum_{i \in B} a_{ij}e_i \right) - H \left( x - \sum_{i \in B} a_{ij}e_i - a_{ij}e_\ell \right)
$$

$$
= H(x) - H \left( x - \sum_{i \in B} a_{ij}e_i \right) + H(w) - H(w - a_{ij}e_\ell)
$$

$$
\leq \Delta_{\mathcal{B}} \sum_{i \in B} \frac{2m_i - 1}{C_i} \sum_{k \in \mathcal{J}} \mathbb{1}_{\{x \in A^k\}} \gamma_k + \sum_{k \in \mathcal{J}} \mathbb{1}_{\{\ell \in A^k\}} \gamma_k
$$

$$
\leq \Delta_{\mathcal{B}} \sum_{i \in B} \frac{2m_i - 1}{C_i} \sum_{k \in \mathcal{J}} \mathbb{1}_{\{x \in A^k\}} \gamma_k + \frac{\Delta_{\mathcal{B}}(2m_\ell - 1)}{C_\ell} \sum_{k \in \mathcal{J}} \mathbb{1}_{\{x \in A^k\}} \gamma_k
$$

$$
= \Delta_{\mathcal{B}} \sum_{i \in A^j} \frac{2m_i - 1}{C_i} \sum_{k \in \mathcal{J}} \mathbb{1}_{\{x \in A^k\}} \gamma_k,
$$

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. Q.E.D.

The next lemma gives an upper bound on the optimal total expected revenue and it generalizes Lemma 3.8 to the case where each product can use multiple units of a resource.

**Lemma B.2 (Upper Bound on Optimal Total Expected Revenue)** If the value function approximations $\{H^t : t \in \mathcal{T}\}$ are constructed using (11), then $V^1(C) \leq (1 + \theta(2M - 1)L)H^1(C)$.

**Proof:** Considering the linear program in the proof of Lemma 3.8, if we 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 $\sum_{i \in A^j} e_i$ with $\sum_{i \in A^j} a_{ij}e_i$ and solve this linear program, then the optimal objective value provides the optimal total expected revenue. Thus, since the linear program minimizes its objective function, the objective value of the linear program at any feasible solution is an upper bound on the optimal total expected revenue $V^1(C)$. We define
the solution \( \{ \nu^t(\mathbf{x}) : \mathbf{x} \in \mathcal{Q}, \ t \in \mathcal{T} \} \) to the linear program as \( \nu^t(\mathbf{x}) = \alpha^t + \sum_{i \in \mathcal{L}} \beta^t_i x_i \), where the coefficients \( \{ \alpha^t : t \in \mathcal{T} \} \) and \( \{ \beta^t_i : i \in \mathcal{L}, \ t \in \mathcal{T} \} \) are given by

\[
\alpha^t = \sum_{j \in \mathcal{J}} \gamma^t_j \quad \text{and} \quad \beta^t_i = \frac{\theta (2m_i - 1)}{C_i} \sum_{k \in \mathcal{J}} 1_{\{i \in A^k\}} \gamma^t_k.
\]

We show that \( \{ \nu^t(\mathbf{x}) : \mathbf{x} \in \mathcal{Q}, \ t \in \mathcal{T} \} \) is a feasible solution to the linear program in the proof of Lemma 3.8 after replacing \( \prod_{i \in A^j} \mathbb{I}_{\{x_i \geq 1\}} \) with \( \prod_{i \in A^j} \mathbb{I}_{\{x_i \geq a_{ij}\}} \) and \( \sum_{i \in A^j} e_i \) with \( \sum_{i \in A^j} a_{ij} e_i \).

To show the feasibility of the solution \( \{ \nu^t(\mathbf{x}) : \mathbf{x} \in \mathcal{Q}, \ t \in \mathcal{T} \} \), computing the right side of the constraint in the linear program at this solution, we get

\[
\nu^{t+1}(\mathbf{x}) + \sum_{j \in \mathcal{J}} \lambda^t_j \left( \prod_{i \in A^j} 1_{\{x_i \geq a_{ij}\}} \right) \left[ r_j - \nu^{t+1}(\mathbf{x}) + \nu^{t+1} \left( \mathbf{x} - \sum_{i \in A^j} a_{ij} e_i \right) \right]^+ \\
= \sum_{j \in \mathcal{J}} \gamma^t_j + \sum_{i \in \mathcal{L}} \beta^t_i x_i + \sum_{j \in \mathcal{J}} \lambda^t_j \left( \prod_{i \in A^j} 1_{\{x_i \geq a_{ij}\}} \right) \left[ r_j - \sum_{i \in A^j} a_{ij} \beta^t_i \right]^+ \\
\leq \sum_{j \in \mathcal{J}} \gamma^t_j + \sum_{i \in \mathcal{L}} \beta^t_i x_i + \sum_{j \in \mathcal{J}} \lambda^t_j \left( \prod_{i \in A^j} 1_{\{x_i \geq a_{ij}\}} \right) \left[ r_j - \sum_{i \in A^j} \beta^t_i \right]^+ \\
= \sum_{j \in \mathcal{J}} \gamma^t_j + \sum_{i \in \mathcal{L}} \beta^t_i x_i + \sum_{j \in \mathcal{J}} \left( \prod_{i \in A^j} 1_{\{x_i \geq a_{ij}\}} \right) (\gamma^t_j - \gamma^t_j) \\
\leq \sum_{j \in \mathcal{J}} \gamma^t_j + \sum_{i \in \mathcal{L}} \beta^t_i x_i + \sum_{j \in \mathcal{J}} (\gamma^t_j - \gamma^t_j) \leq \sum_{j \in \mathcal{J}} \gamma^t_j + \sum_{i \in \mathcal{L}} \beta^t_i x_i = \nu^t(\mathbf{x}),
\]

where the first inequality follows from \( \beta^t_i \geq 0 \) and \( a_{ij} \geq 1 \) for all \( i \in A^j \), the third equality uses (11), and the last two inequalities follow from (11), which implies that \( \gamma^t_j \geq \gamma^t_j \), and thus, \( \beta^t_i \geq \beta^t_i \).

By the chain of inequalities above, the solution \( \{ \nu^t(\mathbf{x}) : \mathbf{x} \in \mathcal{Q}, \ t \in \mathcal{T} \} \) is feasible to the linear program. Hence, the objective value at this solution is an upper bound on \( V^1(C) \), so we get

\[
V^1(C) \leq \nu^1(C) = \alpha^1 + \sum_{i \in \mathcal{L}} \beta^1_i C_i = \sum_{j \in \mathcal{J}} \gamma^1_j + \theta \sum_{i \in \mathcal{L}} (2m_i - 1) \sum_{k \in \mathcal{J}} 1_{\{i \in A^k\}} \gamma^1_k \\
= \sum_{j \in \mathcal{J}} \gamma^1_j + \theta \sum_{k \in \mathcal{J}} \gamma^1_k \sum_{i \in \mathcal{L}} (2m_i - 1) \leq \sum_{j \in \mathcal{J}} \gamma^1_j + \theta (2M - 1) \sum_{k \in \mathcal{J}} \gamma^1_k |A^k| \\
\leq (1 + \theta (2M - 1)L) \sum_{j \in \mathcal{J}} \gamma^1_j = (1 + \theta (2M - 1)L) H^1(C),
\]

where the second and third inequalities are by the definitions of \( M \) and \( L \), whereas the last equality holds because \( \sum_{j \in \mathcal{J}} \gamma^1_j = \sum_{j \in \mathcal{J}} \gamma^1_j f_{A^j}(C) = H^1(C) \) by the third property in Definition 3.1. \( \square \)

As discussed at the beginning of this appendix, if \( r_j \geq V^{t+1}(\mathbf{x}) - V^{t+1}(\mathbf{x} - \sum_{i \in A^j} a_{ij} e_i) \), then it is optimal to accept a request for product \( j \), as long as we have \( \prod_{i \in A^j} \mathbb{I}_{\{x_i \geq a_{ij}\}} = 1 \). Replacing
Letting $U^t(x)$ be the total expected revenue of the approximate policy over time periods $t, t+1, \ldots, T$ given that the system is in state $x$ at time period $t$, we have the recursion

$$U^t(x) = \sum_{j \in J} \lambda_j^t u^{\text{App},t}_j(x) \left[ r_j + U^{t+1} \left( x - \sum_{i \in A} a_{ij} e_i \right) \right] + \left( 1 - \sum_{j \in J} \lambda_j^t + \sum_{j \in J} \lambda_j^t (1 - u^{\text{App},t}_j(x)) \right) U^{t+1}(x)$$

$$= U^{t+1}(x) + \sum_{j \in J} \lambda_j^t u^{\text{App},t}_j(x) \left[ r_j - U^{t+1}(x) + U^{t+1} \left( x - \sum_{i \in A} a_{ij} e_i \right) \right],$$

with the boundary condition that $U^{T+1} = 0$. The recursion above is the analogue of the recursion in (5) for the case where each product can use multiple units of a resource.

Now, here is the proof of Theorem 4.3.

**Proof of Theorem 4.3:** We use induction over the time periods to show that $U^t(x) \geq H^t(x)$ for all $x \in Q$ and $t \in T$. Since $U^{T+1} = H^{T+1} = 0$, the result holds at time period $T+1$. Suppose that $U^{t+1} \geq H^{t+1}$. Replacing $U^{t+1}$ on the right side of (17) with $H^{t+1}$, we obtain

$$U^t(x) \geq H^{t+1}(x) + \sum_{j \in J} \lambda_j^t u^{\text{App},t}_j(x) \left[ r_j - H^{t+1}(x) + H^{t+1} \left( x - \sum_{i \in A} a_{ij} e_i \right) \right]$$

$$= H^{t+1}(x) + \sum_{j \in J} \lambda_j^t \left( \prod_{i \in A} 1_{\{x_i \geq a_{ij}\}} \right) \left[ r_j - H^{t+1}(x) + H^{t+1} \left( x - \sum_{i \in A} a_{ij} e_i \right) \right]$$

$$\geq H^{t+1}(x) + \sum_{j \in J} \lambda_j^t \left( \prod_{i \in A} 1_{\{x_i \geq a_{ij}\}} \right) \left[ r_j - \Delta \sum_{i \in A} \sum_{k \in J} \frac{2m_i - 1}{C_i} \sum_{k \in J} 1_{\{x_k \geq \gamma^{t+1}_k\}} \right]$$

$$\geq H^{t+1}(x) + \sum_{j \in J} \lambda_j^t \left( \prod_{i \in A} 1_{\{x_i \geq a_{ij}\}} \right) \left[ r_j - \theta \sum_{i \in A} \sum_{k \in J} \frac{2m_i - 1}{C_i} \sum_{k \in J} 1_{\{x_k \geq \gamma^{t+1}_k\}} \right]$$

$$= H^{t+1}(x) + \sum_{j \in J} \varphi_{A_j}(g_j(x)) \left( \gamma^{t}_j - \gamma^{t+1}_j \right)$$

$$\geq H^{t+1}(x) + \sum_{j \in J} \varphi_{A_j}(g_j(x)) \left( \gamma^{t}_j - \gamma^{t+1}_j \right)$$

$$= H^t(x),$$

where the first equality holds by the definition of the decision function for the approximate policy given in (16). The second inequality follows from Lemma B.1. The third inequality holds because we choose the tuning parameter $\theta$ to satisfy $\theta \geq \Delta$. The second equality follows from (11). To see
that the fourth inequality holds, note that we have \( \varphi_{A_j}(G_j(x)) \leq \prod_{i \in A_j} \mathbb{1}_{\{x_i \geq a_{ij}\}} \). In particular, if \( \prod_{i \in A_j} \mathbb{1}_{\{x_i \geq a_{ij}\}} = 1 \), then the last inequality holds trivially since \( \varphi_A : \mathcal{Q} \to [0, 1] \) for all \( A \subseteq \mathcal{L} \). If, on the other hand, \( \prod_{i \in A_j} \mathbb{1}_{\{x_i \geq a_{ij}\}} = 0 \), then there must exist some \( i \in \mathcal{L} \) such that \( x_i < a_{ij} \), in which case, the \( i^{th} \) component of \( G_j(x) \) will be zero by the definition of \( G_j \), so by the first property in Definition 3.1, we get \( \varphi_{A_j}(G_j(x)) = 0 \). Therefore, the fourth inequality holds because we have \( \varphi_{A_j}(G_j(x)) \leq \prod_{i \in A_j} \mathbb{1}_{\{x_i \geq a_{ij}\}} \) and (11) implies that \( \gamma^*_j - \gamma^{t+1}_j \geq 0 \). The last equality holds because \( H^{t+1}(x) = \sum_{j \in \mathcal{J}} \gamma^{t+1}_j \varphi_{A_j}(G_j(x)) \). The chain of inequalities above completes the induction argument, so it follows that \( U^t(x) \geq H^t(x) \) for all \( x \in \mathcal{Q} \) and \( t \in \mathcal{T} \).

Noting Lemma B.2 and using the inequality at the end of the previous paragraph with \( t = 1 \) and \( x = C \), we get \((1 + \theta (2M - 1)L) U^1(C) \geq (1 + \theta (2M - 1)L) H^1(C) \geq V^1(C)\), as desired. \( \square \)

Appendix C: Proof of Theorem 4.4

The next lemma shows that we can use our value function approximations to get an upper bound on the optimal objective value of the linear programming approximation.

**Lemma C.1** Letting \( Z^{\text{LP}} \) be the optimal objective value of the linear programming approximation in (12), we have \( Z^{\text{LP}} \leq (1 + \theta L) H^1(C) \).

**Proof:** Noting that \( \varphi_A(C) = 1 \) for all \( A \subseteq \mathcal{L} \) by the third property in Definition 3.1, since we have \( H^{t+1} = 0 \), a telescoping sum yields

\[
H^1(C) = \sum_{i \in \mathcal{T}} (H^i(C) - H^{i+1}(C)) = \sum_{i \in \mathcal{T}} \sum_{j \in \mathcal{J}} (\gamma^*_j - \gamma^{t+1}_j) \varphi_{A_j}(C)
\]

\[
= \sum_{i \in \mathcal{T}} \sum_{j \in \mathcal{J}} \sum_{i \in A_j} \frac{z^*_j \lambda^t_j}{C_j} \left[ r_j - \theta \sum_{i \in A_j} \frac{1}{C_i} \sum_{k \in \mathcal{J}} \mathbb{1}_{\{i \in A_k\}} \gamma^t_k \right]^+
\]

\[
\geq Z^{\text{LP}} - \sum_{i \in \mathcal{T}} \sum_{j \in \mathcal{J}} \varphi_{A_j}(C) \frac{1}{C_j} \sum_{k \in \mathcal{J}} \mathbb{1}_{\{i \in A_k\}} \gamma_k^{t+1}
\]

\[
\geq Z^{\text{LP}} - \sum_{j \in \mathcal{J}} \varphi_{A_j}(C) \frac{1}{C_j} \sum_{k \in \mathcal{J}} \sum_{i \in \mathcal{L}} \mathbb{1}_{\{i \in A_k\}} \gamma_k^1
\]

\[
= Z^{\text{LP}} - \theta \sum_{i \in \mathcal{L}} \sum_{j \in \mathcal{J}} \mathbb{1}_{\{i \in A_j\}} \frac{1}{C_j} \sum_{k \in \mathcal{J}} \mathbb{1}_{\{i \in A_k\}} \gamma_k^1
\]

\[
= Z^{\text{LP}} - \theta \sum_{i \in \mathcal{L}} \sum_{k \in \mathcal{J}} \mathbb{1}_{\{i \in A_k\}} \gamma_k^1 \geq Z^{\text{LP}} - \theta L \sum_{k \in \mathcal{J}} \gamma_k^1 = Z^{\text{LP}} - \theta L H^1(C).
\]

In the chain of inequalities above, the third equality follows from (13). The first inequality follows because \( \sum_{i \in \mathcal{T}} \lambda^t_j = \Lambda_j \), \( Z^{\text{LP}} = \sum_{j \in \mathcal{J}} r_j z^*_j \), and \( \lceil a \rceil^+ \geq a \). The second inequality holds
because $\gamma_j^1 \geq \gamma_j^2 \geq \ldots \geq \gamma_j^{T+1}$ by (13). The fourth equality uses the fact that $i \in A^j$ if and only if $1_{i \in A^j} = 1$. The third inequality holds because the first constraint in the linear programming approximation implies that $\sum_{j \in J} 1_{i \in A^j} z_j^* \leq C_i$. The fourth inequality uses the fact that $\sum_{i \in A^k} 1_{i \in A^k} = |A^k| \leq L$. The last equality is by the fact that $\varphi_{A^k}(C) = 1$ by the third property in Definition 3.1. The chain of inequalities above implies that $(1 + \theta L) H^1(C) \geq Z^{LP}$, as desired. □

The decision function of our approximate policy has the same form as the one in (4). The only difference is that we compute the coefficients $\{\gamma_j^t : j \in J, \ t \in T\}$ by using (13). Furthermore, we can compute the total expected revenue obtained by our approximate policy by using the recursion in (5). The only difference is that the decision function in this recursion needs to correspond to the one computed by using (13). Here is the proof of Theorem 4.4.

**Proof of Theorem 4.4:** We use induction over the time periods to show that $U^t(x) \geq H^t(x)$ for all $x \in Q$ and $t \in T$. The base case corresponding to $t = T + 1$ holds because $U^{T+1} = H^{T+1} = 0$. Suppose that $U^{t+1} \geq H^{t+1}$. Replacing $U^{t+1}$ on the right side of (5) with $H^{t+1}$, the right side of (5) becomes smaller, so we get

$$U^t(x) \geq H^{t+1}(x) + \sum_{j \in J} \lambda_j^t u_j^{app,t}(x) \left[ r_j - H^{t+1}(x) + H^{t+1} \left( x - \sum_{i \in A^j} e_i \right) \right]$$

$$= H^{t+1}(x) + \sum_{j \in J} \lambda_j^t \left( \prod_{i \in A^j} 1_{\{i \geq 1\}} \right) \left[ r_j - H^{t+1}(x) + H^{t+1} \left( x - \sum_{i \in A^j} e_i \right) \right]$$

$$\geq H^{t+1}(x) + \sum_{j \in J} \lambda_j^t \left( \prod_{i \in A^j} 1_{\{i \geq 1\}} \right) \left[ r_j - \theta \sum_{i \in A^j} \frac{1}{C_i} \sum_{k \in J} 1_{\{i \in A^k\}} \gamma_k^{t+1} \right]$$

$$\geq H^{t+1}(x) + \sum_{j \in J} \frac{z_j^t}{\lambda_j^t} \lambda_j^t \left( \prod_{i \in A^j} 1_{\{i \geq 1\}} \right) \left( \gamma_j^t - \gamma_j^{t+1} \right)$$

$$= H^{t+1}(x) + \sum_{j \in J} \varphi_{A^j}(x) \left( \gamma_j^t - \gamma_j^{t+1} \right) = H^t(x),$$

where we use the same argument used in the chain of inequalities in the proof of Theorem 3.6 in the main text. The only difference is that the third inequality uses the fact that $z_j^t \leq \Lambda_j$ by the second constraint in problem (12), along with the fact that $\theta \geq \Delta_{\varphi}$. The chain of inequalities above completes the induction argument so that $U^t(x) \geq H^t(x)$ for all $x \in Q$ and $t \in T$.

Using the inequality at the end of the previous paragraph with $t = 1$ and $x = C$, we have $U^1(C) \geq H^1(C)$. The optimal objective value of the linear programming approximation is an upper bound on the optimal total expected revenue, so $Z^{LP} \geq V^1(C)$; see Bertsimas and Popescu (2003). In this case, noting Lemma C.1, we get $U^1(C) \geq H^1(C) \geq Z^{LP}/(1 + \theta L) \geq V^1(C)/(1 + \theta L)$. □