Performance Guarantees for Network Revenue Management with Flexible Products

Wenchang Zhu, Huseyin Topaloglu
School of Operations Research and Information Engineering, Cornell Tech, New York, NY 10044, USA
wz368@cornell.edu, topaloglu@orie.cornell.edu

July 21, 2023

Problem Definition: We consider network revenue management problems with flexible products. We have a network of resources with limited capacities. To each customer arriving into the system, we offer an assortment of products. The customer chooses a product within the offered assortment or decides to leave without a purchase. The products are flexible in the sense that there are multiple possible combinations of resources that we can use to serve a customer with a purchase for a particular product. We refer to each such combination of resources as a route. The service provider chooses the route to serve a customer with a purchase for a particular product. Such flexible products occur, for example, when customers book at-home cleaning services, but leave the timing of service to the company that provides the service. Our goal is to find a policy to decide which assortment of products to offer to each customer to maximize the total expected revenue, while making sure that there are always feasible route assignments for the customers with purchased products. *Methodology and Results*: We start by considering the case where we make the route assignments at the end of the selling horizon. The dynamic programming formulation of the problem is significantly different from its analogue without flexible products, as the state variable keeps track of the number of purchases for each product, rather than the remaining capacity of each resource. Letting L be the maximum number of resources in a route, we give a policy that obtains at least $1/(1+L)$ fraction of the optimal total expected revenue. We extend our policy to the case where we make the route assignments periodically over the selling horizon. *Managerial Implications*: To our knowledge, the policy that we develop is the first with a performance guarantee under flexible products. Thus, our work constructs policies that can be implemented in practice under flexible products, while also providing performance guarantees.

1. Introduction

In network revenue management problems, we manage the limited capacities of resources to serve the demand for products that arrive randomly over time. These problems find applications in areas as diverse as air travel, hospitality, media advertising and cloud computing. In air travel, for example, the resources take the form of capacities on the flight legs, whereas the products take the form of itineraries that can potentially use multiple flight legs. In the traditional network revenue management setting, the products are inflexible in the sense that the sale of a product consumes the capacities of a fixed combination of resources that depends on the product. The sale of a ticket for a particular itinerary, for example, consumes the capacities of the flight legs in the itinerary. Over the last decade or so, partly fueled by the larger role that online retail and the sharing economy started playing in our lives and partly fueled by the need of service providers to utilize their resources more effectively, offering flexible products became more common. In network revenue management problems with flexible products, there are multiple possible combinations of resources that we can use to serve a customer with a purchase for a particular product. There are many examples of flexible products in air travel, hospitality, online retail and the sharing economy. Airlines sell discount tickets for a specific origin-destination pair and departure date, but do not specify the exact itinerary the customer will take until a day or two before the departure time. Online travel agencies sell hotel rooms for a particular star rating and geographic area, but disclose the specific property the customer is assigned to only after the purchase happens. Online retailers sell clothing items but do not specify the color of the item the customer will get in return for a discount. Providers of at-home cleaning or pet walking services allow their customers to specify whether they prefer morning or afternoon hours, but the service provider gets to choose the time of service. Especially in hospitality and online retail, flexible products are often referred to as opaque products. In many applications, flexible and inflexible products are offered together with a premium attached to inflexible ones. In some settings, the service provider needs to decide which combination of resources to use to serve a flexible product purchase right after the purchase happens, which is usually the case for hospitality and online retail. In other settings, the service provider can delay the decision, which is usually the case for airlines or at-home service providers.

We consider network revenue management problems with flexible products. We have a network of resources with limited capacities. To each customer arriving into the system, we offer an assortment of products. The customer chooses a product within the offered assortment or decides to leave without a purchase. The products are flexible in the sense that there are multiple combinations of resources that we can use to serve a customer with a purchase for a particular product. Motivated by airline applications, we refer to each combination of resources that we can use to serve a customer as a route, but our model is general enough to encompass other applications discussed in the previous paragraph. Our goal is to find a policy to decide which assortment of products to offer to each customer to maximize the total expected revenue over the selling horizon, while making sure that we always have routes with enough capacity to accommodate all customers with a purchased product on their hands. We start by considering the case where we make the route assignments at the end of the selling horizon, but we make extensions to the case where allow making route assignments at any frequency throughout the selling horizon, including making the route assignment decision for each customer right after her purchase. Our approach also allows having inflexible products, along with flexible ones. In particular, an inflexible product can be viewed as a flexible product that can be served only through a single possible route.

Contributions. Our main technical contribution is a policy that is guaranteed to obtain at least $1/(1+L)$ fraction of the optimal total expected revenue, where L is the maximum number of resources in any route. The number of resources and products in practical applications can be large, but the number of resources in a route is usually uniformly bounded. In airline applications, for example, the number of flights in an itinerary rarely exceeds two, so $L=2$. Flexible products bring novel challenges because the dynamic programming formulation of the network revenue management problem with flexible products is significantly different from its counterpart with inflexible products. Under flexible products, the state variable keeps the numbers of product purchases, whereas under inflexible products, the state variable keeps the remaining resource capacities. When the value functions are functions of the remaining resource capacities, we can often conjecture that they are concave in the remaining resource capacities, guiding the choice of good value function approximations. When the value functions are functions of the numbers of product purchases, it is not clear how to make similar conjectures. Also, under flexible products, to check whether it is feasible to accept a product request, we need to solve a packing problem to find feasible route assignments for the accepted products, whereas under inflexible products, we can simply check whether there is remaining capacity for each resource used by the product.

Our approach is based on constructing approximations to the value functions. Due to the significantly different nature of the dynamic programming formulation under flexible products, it is not immediately clear how to construct a value function approximation. Considering the case with only inflexible products, letting $\mathcal{L}$ be the set of resources, we can use $\boldsymbol{w}=(w_i : i \in \mathcal{L})$ to capture the state of the remaining resource capacities, where w_i is the remaining capacity of resource i. At time period t in the selling horizon, a reasonable value function approximation is of the form $\hat{\Psi}_t(\boldsymbol{w})$, where we expect $\hat{\Psi}_t(\boldsymbol{w})$ to be monotone increasing and concave in w_i. Monotonicity implies that a larger remaining capacity should yield larger total expected revenue, whereas concavity implies that each incremental unit of remaining capacity should yield smaller marginal total expected revenue. Turning to our case with flexible products, letting $\mathcal{J}$ be the set of products, we can use $\boldsymbol{x}=(x_j : j \in \mathcal{J})$ to capture the state of the product purchases, where x_j is the number of purchases for product j. Letting $\mathcal{W}(\boldsymbol{x})$ be the set of all possible remaining resource capacities after making the route assignment decisions for the product purchases $\boldsymbol{x}$, one of our critical contributions is to use a value function approximation of the form $\hat{J}_t(\boldsymbol{x}) = \max_{\boldsymbol{w}\in\mathcal{W}(\boldsymbol{x})} \hat{\Psi}_t(\boldsymbol{w})$.

Note that simply computing our value function approximation at a particular point requires solving an optimization problem. Intuitively, by using this optimization problem, we *convert* the value function approximation $\hat{\Psi}_t(\boldsymbol{w})$, which is defined as a function of the remaining resource capacities, into the value function approximation $\hat{J}_t(\boldsymbol{x})$, which is defined as a function of the product purchases. The advantage of using the optimization problem to compute our value function approximations is that there is vast literature on network revenue management without flexible products, guiding our choice of the approximation $\hat{\Psi}_t(\boldsymbol{w})$, in which case, we use the optimization problem to convert the value function approximation $\hat{\Psi}_t(\boldsymbol{w})$ to the value function approximation $\hat{J}_t(\boldsymbol{x})$ for our problem. Such a conversion idea does not appear in the literature to obtain performance guarantees. The implicit assumption in using the optimization problem is that even if we make the route assignment decisions at the end of the selling horizon, the total expected revenue starting at time period t with product purchases $\boldsymbol{x}$ can be approximated by making the route assignment decisions immediately to obtain the remaining resource capacities $\boldsymbol{w}$ and focusing on the total expected revenue starting at time period t with the remaining resource capacities $\boldsymbol{w}$. This assumption turns out to be adequate to get our performance guarantee.

There is work constructing value function approximations without flexible products. Ma et al. (2020) use the so-called availability tracking value function approximations for problems without flexible products. These approximations are functions of the remaining resource capacities. They have one component for each product. The component that corresponds to a certain product takes value zero when there is not enough remaining capacity on some resource to serve the product. The value function approximation $\hat{\Psi}_t(\boldsymbol{w})$ that we use in our optimization problem is also an availability tracking approximation. We give an algorithm to calibrate the parameters of the value function approximation $\hat{\Psi}_t(\boldsymbol{w})$ such that the greedy policy with respect to the value function approximation $\hat{J}_t(\boldsymbol{x}) = \max_{\boldsymbol{w} \in \mathcal{W}(\boldsymbol{x})} \hat{\Psi}_t(\boldsymbol{w})$ has a performance guarantee. In that sense, our work is a generalization of Ma et al. (2020) to flexible products, but it is not a priori clear that we can obtain a performance guarantee under flexible products after we distort the approximation $\hat{\Psi}_t(\boldsymbol{w})$ through the problem $\hat{J}_t(\boldsymbol{x}) = \max_{\boldsymbol{w} \in \mathcal{W}(\boldsymbol{x})} \hat{\Psi}_t(\boldsymbol{w})$. The idea of using the last optimization problem to convert an approximation that is a function of remaining resource capacities to an approximation that is a function of product purchases is the key original driver of our work and we believe that it may find applications in other settings. Moreover, as far as we are aware, there are no available performance guarantees for network revenue management with flexible products.

We show that our approach extends to the case where we periodically make the route assignment decisions without waiting for the end of the selling horizon. We get the same performance guarantee of $1/(1+L)$. This extension is not immediate either because the state variable under periodic route assignments needs to keep track of the resource capacities consumed by the product purchases for which we already made the route assignments, as well as the number of purchases for each product for which we have not yet made the route assignments. Also, we show that we can use our approach to incorporate flexible products into problem settings involving bipartite matching, pricing or reusable resources, further extending the reach of our approach. In our computational experiments, we demonstrate that our policies perform well and we investigate the benefit from the making route assignments with different frequencies. Frequent route assignments are customer-centric as the customers get to know their flight itinerary, hotel property, product color or time of service early on, but delayed route assignments are firm-centric, allowing the service provider to utilize the resources more efficiently. Lastly, there are two sources of difficulty for network revenue management with flexible products. First, the state variable is a high-dimensional vector. Second, when offering a product, we always need to check that we have routes with enough capacity to accommodate all customers with a purchased product. This check needs to be made by any policy, as it is an inherent part of dealing with flexible products. Carrying out this check is NP-complete and we rely on the strength of integer programming solvers for this check, but our approach fully addresses the source of difficulty due to the high-dimensional state variable.

Related Literature. There are a number of papers on revenue management problems with flexible products. Gallego and Phillips (2004) give a stylized model with two flights between an origin-destination pair and one flexible product, allowing the airline to assign the customers to either of the flight legs. Gallego et al. (2004) study a linear programming approximation for network revenue management with flexible products and give policies that are asymptotically optimal as the resource capacities and expected demand get large. Gonsch et al. (2014) build on the linear programming approximation to give heuristic policies. Cheung and Simchi-Levi (2016) focus on solving the linear programming approximation in an approximate fashion. Koch et al. (2017) give a characterization of when it would be optimal to make the route assignment of a customer as soon as the purchase occurs without necessarily waiting for the end of the selling horizon.

Upgradeable products are a form of flexible products, because the service provider may serve a customer with a premium product when the originally purchased product is not available. Shumsky and Zhang (2009) study the structure of the optimal policy when the customers can be upgraded only one level above their original purchase. Gallego and Stefanescu (2009) use a linear programming approximation to manage upgrades. Xu et al. (2011) consider a setting where the customers decide whether to accept a substitute product and establish the concavity of the value functions. Steinhardt and Gonsch (2012) give heuristic policies for managing upgrades, as well as conditions under which it would be optimal to upgrade the customer as soon as the purchase occurs, without waiting for the end of the selling horizon. Yu et al. (2015) characterize the structure of the optimal policy under upgrades with multiple levels.

There is work on pricing and assortment optimization for opaque products. Fay and Xie (2008) use a one-period model to quantify the benefit from offering an opaque product. Xiao and Chen (2014) formulate a dynamic program for selling an opaque product and give upper and lower bounds on the value functions. Fay and Xie (2015) use a two-period model to compare the implications of making the route assignment decisions after the demand uncertainty resolves to different extents. Elmachtoub et al. (2015) study the optimal inventory and allocation policies in the presence of opaque products. The papers so far in this paragraph focus on two resources, using which one opaque product is offered. Elmachtoub et al. (2019) study the design of opaque products. Elmachtoub and Hamilton (2021) use a single-period model to understand when offering opaque products can make up for the expected revenues obtained by other pricing mechanisms that may be perceived as unfair. Overbooking problems also resemble managing flexible products because cancellations prevent the service provider from knowing which resource capacities will be used, so their dynamic programming formulations keep track of the numbers of different product purchases, rather than remaining resource capacities. Bertsimas and Popescu (2003) use a linear programming approximation to make overbooking decisions over a network. Karaesmen and van Ryzin (2004) study an overbooking problem over multiple flights between the same origin-destination pair, where the excess demand from one flight can be shifted to another. Erdelyi and Topaloglu (2010) heuristically decompose the overbooking problem over a flight network by resources.

Ma et al. (2020) establish the performance guarantee of $1/(1+L)$ without flexible products, but as discussed earlier, due to the significantly different nature of the dynamic program under flexible products, it is not clear how to extend this work to flexible products. In particular, the form of the value functions under flexible products is not clear. As far as we are aware, our work is first to provide performance guarantees under flexible products. Also, our problem is a stochastic version of the set packing problem. Hazan et al. (2006) show that it is NP-hard to approximate a set packing problem within a factor of $\Omega(\log L/L)$, where L is the maximum number of elements in a set. Thus, our performance guarantee is accurate up to a logarithmic factor in L. Baek and Ma (2022) extend the $1/(1+L)$ performance guarantee to the case where some of the resource constraints have a matroid structure and the performance guarantee is independent of the constraints in the matroid structure. There have been a number of recent papers on developing policies with performance guarantees for revenue management problems, but these papers do not consider flexible products or network of resources; see, for example, Alaei et al. (2012), Rusmevichientong et al. (2020), Ma et al. (2021), Manshadi and Rodilitz (2022) and Feng et al. (2022).

Organization. In Section 2, we give a dynamic programming formulation under flexible products. In Section 3, we formulate the optimization problem that we solve to compute our value function approximations and give an algorithm to calibrate the parameters of our approximations. In Section 4, we give our policy with a $1/(1+L)$ performance guarantee. In Section 5, we prove the performance guarantee. In Section 6, we extend our work to periodic route assignments. In Section 7, we discuss the links between the performance guarantees under different route assignment frequencies. In Section 8, we give computational experiments. In Section 9, we conclude.

2. Problem Formulation

The set of resources is $\mathcal{L}$. We have c_i units of resource i. The set of products is $\mathcal{J}$. We use f_j to denote the revenue associated with product j. There are T time periods in the selling horizon indexed by $\mathcal{T} = \{1, \ldots, T\}$. The time periods correspond to small enough durations of time that there is one customer arrival at each time period. If we offer the assortment of products $S \subseteq \mathcal{J}$ at time period t, then the arriving customer purchases product j with probability $\phi_{jt}(S)$. With probability $1 - \sum_{j\in\mathcal{J}} \phi_{jt}(S)$, the arriving customer leaves without making a purchase. We assume that the choice probabilities satisfy the substitutability property $\phi_{jt}(S) \geq \phi_{jt}(Q)$ for all $S \subseteq Q \subseteq \mathcal{J}$, $j \in S$ and $t \in \mathcal{T}$, which implies that if we offer a smaller assortment, then the choice probability of each product in the smaller assortment gets larger. All choice models based on random utility maximization satisfy the substitutability property. We use $\mathcal{R}_j$ to denote the set of possible routes to serve a customer with a purchase for product j. Each route corresponds to a combination of resources. To capture the resources used by a route, let $a_{ip} = 1$ if route p uses resource i; otherwise, $a_{ip} = 0$. We make all of the route assignments at the end of the selling horizon.

Our goal is to find a policy to decide which assortment of products to offer at each time period so that we maximize the total expected revenue over the selling horizon, while ensuring that we always have routes with enough capacity to accommodate all of the customers with a purchased product. We proceed to giving a dynamic program to compute the optimal policy. Letting x_j be the number of customers with a purchase for product j at the beginning of a generic time period, we use the vector $\boldsymbol{x} = (x_j : j \in \mathcal{J}) \in \mathbb{Z}_+^{|\mathcal{J}|}$ to capture the state of the system. To ensure that we have routes with enough capacity for all of the customers with a purchased product, we always need to be able to assign each customer with a purchased product to a route in such a way that the route assignments do not violate the resource capacities. To characterize the possible route assignments for the customers, we use the decision variables $\boldsymbol{y} = (y_{jp} : j \in \mathcal{J},\ p \in \mathcal{R}_j) \in \mathbb{Z}_+^{\sum_{j\in\mathcal{J}} |\mathcal{R}_j|}$, where y_{jp} is the number of customers with a purchase for product j that we assign to route p. Thus, if the numbers of customers with purchases for different products are given by the state vector $\boldsymbol{x}$, then we can capture the set of feasible route assignments for the customers as

$$\mathcal{F}(\boldsymbol{x}) = \left\{ \boldsymbol{y} \in \mathbb{Z}_+^{\sum_{j\in\mathcal{J}} |\mathcal{R}_j|} : \sum_{p\in\mathcal{R}_j} y_{jp} = x_j \quad \forall j \in \mathcal{J},\ \sum_{j\in\mathcal{J}} \sum_{p\in\mathcal{R}_j} a_{ip}\, y_{jp} \leq c_i \quad \forall i \in \mathcal{L} \right\}, \tag{1}$$

where the first constraint ensures that we assign each customer to a route and the second constraint ensures that the route assignments do not violate the resource capacities.

Given that the system is in state $\boldsymbol{x}$, if $\mathcal{F}(\boldsymbol{x}) \neq \varnothing$, then there exists a way of making route assignments without violating the capacities of the resources. In this case, using $\boldsymbol{e}_j \in \mathbb{Z}_+^{|\mathcal{J}|}$ to denote the j-th unit vector and $\mathbf{1}_{(\cdot)}$ to denote the indicator function, we can find the optimal policy by computing the value functions $\{J_t : t \in \mathcal{T}\}$ through the dynamic program

$$\begin{aligned} J_t(\boldsymbol{x}) &= \max_{S \subseteq \mathcal{J}} \left\{ \sum_{j \in \mathcal{J}} \phi_{jt}(S)\, \mathbf{1}_{(\mathcal{F}(\boldsymbol{x}+\boldsymbol{e}_j) \neq \varnothing)} \Big[f_j + J_{t+1}(\boldsymbol{x}+\boldsymbol{e}_j) \Big] + \Big[1 - \sum_{j \in S} \phi_{jt}(S)\, F(\boldsymbol{x}+\boldsymbol{e}_j) \Big] J_{t+1}(\boldsymbol{x}) \right\} \\ &= \max_{S \subseteq \mathcal{J}} \left\{ \sum_{j \in \mathcal{J}} \phi_{jt}(S)\, \mathbf{1}_{(\mathcal{F}(\boldsymbol{x}+\boldsymbol{e}_j) \neq \varnothing)} \Big[f_j + J_{t+1}(\boldsymbol{x}+\boldsymbol{e}_j) - J_{t+1}(\boldsymbol{x}) \Big] \right\} + J_{t+1}(\boldsymbol{x}), \qquad (2) \end{aligned}$$

with the boundary condition $J_{T+1}(\boldsymbol{x}) = 0$ for all $\boldsymbol{x} \in \mathbb{Z}_+^{|\mathcal{J}|}$. In the first equality, if a customer chooses product j and there exist feasible route assignments after the purchase, then we generate a revenue of f_j and have one more purchase for product j at the next time period. In the dynamic program above, we can offer product j even if $\mathcal{F}(\boldsymbol{x}+\boldsymbol{e}_j) = \varnothing$ so that there are no feasible route assignments after the purchase for product j, but we argue that there exist an optimal policy that does not offer product j whenever $\mathcal{F}(\boldsymbol{x}+\boldsymbol{e}_j) = \varnothing$. In the maximization problem on the right side of (2), the net revenue contribution of product j is $\mathbf{1}_{(\mathcal{F}(\boldsymbol{x}+\boldsymbol{e}_j) \neq \varnothing)}\,(f_j + J_{t+1}(\boldsymbol{x}+\boldsymbol{e}_j) - J_{t+1}(\boldsymbol{x}))$. In an optimal solution to this problem, if we drop each product j with $\mathbf{1}_{(\mathcal{F}(\boldsymbol{x}+\boldsymbol{e}_j) \neq \varnothing)}\,(f_j + J_{t+1}(\boldsymbol{x}+\boldsymbol{e}_j) - J_{t+1}(\boldsymbol{x})) \leq 0$, then by the substitutability property, the choice probability of all other products in the solution increases. In this way, we eliminate all products with non-positive net revenue contributions from the optimal solution and the remaining products have larger choice probabilities, so the solution that we obtain must also be optimal. Thus, the dynamic program generates a policy that does not offer product j whenever $\mathcal{F}(\boldsymbol{x}+\boldsymbol{e}_j) = \varnothing$. The substitutability property will be useful once more later in the paper to construct an upper bound on the optimal total expected revenue.

In an airline application, for example, a resource is a flight leg, a product is a reservation for an origin-destination pair and a route is a sequence of connecting flight legs that can be used to serve a reservation. We can serve a reservation for a flexible product using one of several possible routes, whereas we can serve a reservation for an inflexible product using one specific route. Thus, if product j is a flexible product, then $|\mathcal{R}_j| > 1$, whereas if product j is an inflexible product, then $|\mathcal{R}_j| = 1$. We make the decision as to which route to use to serve a reservation for a flexible product just before the departure time. There are two sources of difficulty for the dynamic program in (2). First, the state variable is a high-dimensional vector, so storing the value function $J_t(\boldsymbol{x})$ for each possible state vector $\boldsymbol{x}$ is intractable. Second, computing the value of $\mathbf{1}_{(\mathcal{F}(\boldsymbol{x}) \neq \varnothing)}$ at any state vector $\boldsymbol{x}$ requires finding out whether the set in (1) is non-empty, which is equivalent to checking the feasibility of a packing problem. In Appendix A, we give a reduction from the set packing problem to argue that computing the value of $\mathbf{1}_{(\mathcal{F}(\boldsymbol{x}) \neq \varnothing)}$ is NP-complete. In this paper, our approximation strategy fully addresses the first source of difficulty, but it will require computing the value of $\mathbf{1}_{(\mathcal{F}(\boldsymbol{x}) \neq \varnothing)}$ and we rely on the strength of integer programming solvers for this purpose.

3. Value Function Approximations

We construct an approximation to the value function $J_t(\boldsymbol{x})$. The argument $\boldsymbol{x}$ in this value function keeps track of the numbers of purchases for different products. It is difficult to conjecture a form for a value function approximation when the state variable keeps track of the numbers of purchases for different products. Instead, we start with an auxiliary value function approximation whose argument keeps track of the remaining capacities of the resources. To approximate the value of $J_t(\boldsymbol{x})$ at any vector of product purchases $\boldsymbol{x}$, we make the route assignments for all these product purchases in such a way that we maximize the value of the auxiliary approximation attained at the remaining resource capacities after the route assignments. To formalize, we capture the remaining resource capacities by using the vector $\boldsymbol{w} = (w_i : i \in \mathcal{L})$, where w_i is the remaining capacity for resource i. Given the vector of product purchases $\boldsymbol{x}$ at time period t, if making the route assignments for these product purchases immediately leaves us with the remaining resource capacities $\boldsymbol{w}$, then we approximate the value function at time period t by a function of the form $\hat{\Psi}_t(\boldsymbol{w})$. Thus, our approximation to $J_t(\boldsymbol{x})$ is given by the optimal objective value of the problem

$$\hat{J}_t(\boldsymbol{x}) = \max_{(\boldsymbol{y},\boldsymbol{w}) \in \mathbb{Z}_+^{\sum_{j\in\mathcal{J}} |\mathcal{R}_j| + |\mathcal{L}|}} \Bigg\{ \hat{\Psi}_t(\boldsymbol{w}) \; : \; \sum_{p \in \mathcal{R}_j} y_{jp} = x_j \quad \forall\, j \in \mathcal{J}, \qquad \sum_{j\in\mathcal{J}} \sum_{p\in\mathcal{R}_j} a_{ip}\, y_{jp} + w_i = c_i \quad \forall\, i \in \mathcal{L} \Bigg\}, \tag{3}$$

where the two constraints are similar to those in (1), but the second constraint above explicitly computes the remaining capacity of each resource after route assignments.

Problem (3), intuitively speaking, converts the auxiliary value function approximation $\hat{\Psi}_t(\boldsymbol{w})$ to $\hat{J}_t(\boldsymbol{x})$. Thus, we need to solve problem (3) just to compute the value function approximation $\hat{J}_t(\boldsymbol{x})$ for one value of $\boldsymbol{x}$. Throughout the paper, using $A_p = \{i \in \mathcal{L} : a_{ip} = 1\}$ to denote the set of resources used by route p, letting $\psi_p(\boldsymbol{w}) = \min_{i\in A_p}\{\frac{w_i}{c_i}\}$ for notational brevity, we use the functional form $\hat{\Psi}_t(\boldsymbol{w}) = \sum_{j\in\mathcal{J}} \sum_{p\in\mathcal{R}_j} \hat{\gamma}_{jpt}\, \psi_p(\boldsymbol{w})$, where $(\hat{\gamma}_{jpt} : j \in \mathcal{J},\ p \in \mathcal{R}_j,\ t \in \mathcal{T})$ are adjustable parameters. We shortly give an algorithm to calibrate these parameters. To motivate the form of the approximation $\hat{\Psi}_t(\boldsymbol{w}) = \sum_{j\in\mathcal{J}} \sum_{p\in\mathcal{R}_j} \hat{\gamma}_{jpt}\, \psi_p(\boldsymbol{w})$, recall that if the remaining capacities after we make the route assignments are given by the vector $\boldsymbol{w}$, then we approximate the optimal total expected revenue over the time periods $t, \ldots, T$ by using $\sum_{j\in\mathcal{J}} \sum_{p\in\mathcal{R}_j} \hat{\gamma}_{jpt}\, \psi_p(\boldsymbol{w})$. Using $\boldsymbol{c} = (c_i : i \in \mathcal{L})$ to denote full resource capacities, by the definition of $\psi_p(\boldsymbol{w})$, we have $\psi_p(\boldsymbol{c}) = 1$. Therefore, if we have full capacities, then we approximate the optimal total expected revenue over the time periods $t, \ldots, T$ by $\sum_{j\in\mathcal{J}} \sum_{p\in\mathcal{R}_j} \hat{\gamma}_{jpt}$. We interpret $\hat{\gamma}_{jpt}$ as an approximation to the total expected revenue from the requests for product j that are assigned to route p, given that we have full capacities. Because

$\psi_p(\boldsymbol{w}) \in [0,1]$ for any $\boldsymbol{w} \in \mathbb{Z}_+^{|\mathcal{L}|}$, we modulate the approximation $\hat{\gamma}_{jpt}$ by $\psi_p(\boldsymbol{w})$ depending on the resource availabilities. If $w_i = 0$ for some $i \in A_p$, so that we do not have capacity for a resource in route p, then $\psi_p(\boldsymbol{w}) = 0$. Thus, our approximation to the total expected revenue from the requests for product j that are assigned to route p is zero. Noting that $\psi_p(\boldsymbol{w}) = \min_{i \in A_p}\{\frac{w_i}{c_i}\}$, we can formulate (3) as an integer program by using the additional decision variables $(u_{jp} : j \in \mathcal{J},\ p \in \mathcal{R}_j)$, replacing the objective function with $\sum_{j \in \mathcal{J}} \sum_{p \in \mathcal{R}_j} \hat{\gamma}_{jpt}\, u_{jp}$ and adding the constraints $u_{jp} \leq \frac{w_i}{c_i}$ for all $j \in \mathcal{J}$, $p \in \mathcal{R}_j$ and $i \in A_p$ in (3). To fully specify the approximation $\sum_{j \in \mathcal{J}} \sum_{p \in \mathcal{R}_j} \hat{\gamma}_{jpt}\, \psi_p(\boldsymbol{w})$, we need to calibrate the parameters $(\hat{\gamma}_{jpt} : j \in \mathcal{J},\ p \in \mathcal{R}_j,\ t \in \mathcal{T})$. We use the following algorithm for calibration. We set $\hat{\gamma}_{jp,T+1} = 0$ for all $j \in \mathcal{J}$ and $p \in \mathcal{R}_j$. Letting $\theta \geq 1$ be a tuning parameter, starting from the last time period, for each $t = T, T-1, \ldots, 1$, we execute the three steps.

- Find an ideal route for each product at the current time period: For each $j \in \mathcal{J}$, set the ideal route $\hat{p}_{jt}$ at the current time period as

$$\hat{p}_{jt} \;=\; \arg\max_{p \in \mathcal{R}_j} \left\{ f_j - \theta \sum_{i \in A_p} \frac{1}{c_i} \sum_{k \in \mathcal{J}} \sum_{q \in \mathcal{R}_k} a_{iq}\, \hat{\gamma}_{kq,t+1} \right\}. \tag{4}$$

- Choose the ideal assortment at the current time period: Set the ideal assortment of products $\hat{S}_t$ at the current time period as

$$\hat{S}_t \;=\; \arg\max_{S \subseteq \mathcal{J}} \left\{ \sum_{j \in \mathcal{J}} \phi_{jt}(S) \left(f_j - \theta \sum_{i \in A_{\hat{p}_{jt}}} \frac{1}{c_i} \sum_{k \in \mathcal{J}} \sum_{q \in \mathcal{R}_k} a_{iq}\, \hat{\gamma}_{kq,t+1} \right) \right\}. \tag{5}$$

- Compute the adjustable parameters as the current time period: For each $j \in \mathcal{J}$ and $p \in \mathcal{R}_j$, set the adjustable parameter $\hat{\gamma}_{jpt}$ at the current time period as

$$\hat{\gamma}_{jpt} \;=\; \phi_{jt}(\hat{S}_t)\; \mathbf{1}_{(\hat{p}_{jt} = p)} \left(f_j - \theta \sum_{i \in A_p} \frac{1}{c_i} \sum_{k \in \mathcal{J}} \sum_{q \in \mathcal{R}_k} a_{iq}\, \hat{\gamma}_{kq,t+1} \right) + \hat{\gamma}_{jp,t+1}. \tag{6}$$

The algorithm above specifies the parameters $(\hat{\gamma}_{jpt} : j \in \mathcal{J}\ , p \in \mathcal{R}_j,\ t \in \mathcal{T})$, in which case, we can use (3) to compute the value function approximation $\hat{J}_t(\boldsymbol{x})$ at any $\boldsymbol{x} \in \mathbb{Z}_+^{|\mathcal{J}|}$.

We interpret (4)-(6). Considering $\hat{\Psi}_t(\boldsymbol{w}) = \sum_{j \in \mathcal{J}} \sum_{p \in \mathcal{R}_j} \hat{\gamma}_{jpt}\, \psi_p(\boldsymbol{w})$, using $\overline{\boldsymbol{e}}_i \in \mathbb{Z}_+^{|\mathcal{L}|}$ to denote the i-th unit vector, the difference $\hat{\Psi}_t(\boldsymbol{w}) - \hat{\Psi}_t(\boldsymbol{w} - \sum_{i \in A_p} \overline{\boldsymbol{e}}_i)$ captures the opportunity cost of giving up the capacities in route p. Using the definition of $\psi_p(\boldsymbol{w})$, through simple algebra, we can show that $\hat{\Psi}_t(\boldsymbol{w}) - \hat{\Psi}_t(\boldsymbol{w} - \sum_{i \in A_p} \overline{\boldsymbol{e}}_i) \leq \sum_{i \in A_p} \frac{1}{c_i} \sum_{k \in \mathcal{J}} \sum_{q \in \mathcal{R}_k} a_{iq}\, \hat{\gamma}_{kqt}$. Noting that the right side of the last inequality does not depend on $\boldsymbol{w}$, we use $\sum_{i \in A_p} \frac{1}{c_i} \sum_{k \in \mathcal{J}} \sum_{q \in \mathcal{R}_k} a_{iq}\, \hat{\gamma}_{kqt}$ as a state-independent approximation to the opportunity cost of giving up the capacities in route p. In (4), we compute the ideal route for product j at time period t by finding the route that maximizes the revenue from the product after subtracting the opportunity cost of giving up the capacities in the route. Once we compute the ideal route, we view $f_j - \sum_{i \in A_{\hat{p}_{jt}}} \frac{1}{c_i} \sum_{k \in \mathcal{J}} \sum_{q \in \mathcal{R}_k} a_{iq} \, \hat{\gamma}_{kq,t+1}$ as the net revenue from product j at time period t. In (5), we compute the ideal assortment at time period t by finding the assortment that maximizes the net expected revenue from a customer at time period t. Such ideal assortments appear in Rusmevichientong et al. (2020) and Ma et al. (2020). In (6), recall that $\hat{\gamma}_{jpt}$ captures the total expected revenue over time periods $t, \ldots, T$ from the requests for product j that are assigned to route p. At time period t, if we offer the ideal assortment, then a customer chooses product j with probability $\phi_{jt}(\hat{S}_t)$. If the customer chooses product j, then we assign her to the ideal route $\hat{p}_{jt}$ and get the net revenue $f_j - \sum_{i \in A_{\hat{p}_{jt}}} \frac{1}{c_i} \sum_{k \in \mathcal{J}} \sum_{q \in \mathcal{R}_k} a_{iq} \, \hat{\gamma}_{kq,t+1}$. In addition to the net revenue at time period t, note that $\hat{\gamma}_{jp,t+1}$ on the right side of (6) corresponds to the total expected revenue over time periods $t+1, \ldots, T$.

Letting $L = \max_{j \in \mathcal{J}, \; p \in \mathcal{R}_j} |A_p|$ to capture the maximum number of resources used by any route, for any $\theta \geq 1$, we will show that the greedy policy with respect to the value function approximations $\{\hat{J}_t : t \in \mathcal{T}\}$ is guaranteed to obtain at least $1/(1+\theta L)$ fraction of the optimal total expected revenue. Therefore, setting $\theta = 1$ is enough to come up with a policy that is guaranteed to obtain at least $1/(1+L)$ fraction of the optimal total expected revenue. In our numerical experiments, however, the practical performance of our policy may improve when we use values of θ larger than one, so it may be useful to try values of θ larger than one. Thus, we leave θ as an adjustable parameter, where each choice of θ yields a different policy. In our numerical experiments, we give a systematic approach to choose the value of this parameter.

4. Approximate Policy

We give a description of our approximate policy. Using (4)-(6), we compute the parameters $(\hat{\gamma}_{jpt} : j \in \mathcal{J}, \; p \in \mathcal{R}_j, \; t \in \mathcal{T})$, in which case, we can compute the value function approximation $\hat{J}_t(\boldsymbol{x})$ at any $\boldsymbol{x} \in \mathbb{Z}_+^{|\mathcal{J}|}$ by solving problem (3). Note that computing our value function approximation at a single point requires solving an optimization problem. In our approximate policy, we follow the greedy action with respect to the value function approximations $\{\hat{J}_t : t \in \mathcal{T}\}$. In particular, our approximate policy makes its decisions at time period t by replacing J_{t+1} on the right side of (2) with $\hat{J}_{t+1}$ and solving the corresponding maximization problem. Thus, if the numbers of customers at time period t with purchases for different products are given by the state vector $\boldsymbol{x}$, then our approximate policy offers the assortment of products given by

$$S_t^{\mathsf{App}}(\boldsymbol{x}) \;=\; \max_{S \subseteq \mathcal{J}} \left\{ \sum_{j \in \mathcal{J}} \phi_{jt}(S) \, \mathbf{1}_{(\mathcal{F}(\boldsymbol{x}+\boldsymbol{e}_j) \neq \varnothing)} \left[f_j + \hat{J}_{t+1}(\boldsymbol{x}+\boldsymbol{e}_j) - \hat{J}_{t+1}(\boldsymbol{x}) \right] \right\}. \tag{7}$$

By the same reasoning right after (2), there exists an optimal solution to the problem above such that if $\mathcal{F}(\boldsymbol{x}+\boldsymbol{e}_j) = \varnothing$, then product j is not in $S_t^{\mathsf{App}}(\boldsymbol{x})$. Thus, if there does not exist feasible route assignments after a purchase for product j, then our approximate policy does not offer product j. In the next theorem, we give a performance guarantee for the approximate policy. Recall that $L = \max_{j\in\mathcal{J},\,p\in\mathcal{R}_j} |A_p|$ is the maximum number of resources used by a route.

Theorem 4.1 (Performance Guarantee) *The total expected revenue obtained by the approximate policy is at least* $1/(1+\theta L)$ *fraction of the optimal total expected revenue.*

We give the full proof of the theorem in the next section. We discuss the main ingredients of the proof. We consider a linear program to obtain an upper bound on the optimal total expected revenue. In this linear program, we use the decision variables $(h_t(S) : S \subseteq \mathcal{J},\ t \in \mathcal{T})$, where $h_t(S)$ is the probability of offering assortment S at time period t, as well as the decision variables $(y_{jp} : j \in \mathcal{J},\ p \in \mathcal{R}_j)$, where y_{jp} is the expected number of purchases for product j that we assign to route p. Our linear program can be interpreted as a deterministic approximation to the dynamic program in (2) that is formulated under the assumption that the arrivals and choices of the customers take on their expected values. In particular, we consider the linear program

$$
\begin{aligned}
Z^*_{\mathsf{LP}} \;=\; \max \quad & \sum_{t\in\mathcal{T}} \sum_{S\subseteq\mathcal{J}} \sum_{j\in\mathcal{J}} f_j\, \phi_{jt}(S)\, h_t(S) \\
\text{st} \quad & \sum_{t\in\mathcal{T}} \sum_{S\subseteq\mathcal{J}} \phi_{jt}(S)\, h_t(S) = \sum_{p\in\mathcal{R}_j} y_{jp} \quad \forall\, j\in\mathcal{J} \\
& \sum_{j\in\mathcal{J}} \sum_{p\in\mathcal{R}_j} a_{ip}\, y_{jp} \le c_i \quad \forall\, i\in\mathcal{L} \\
& \sum_{S\subseteq\mathcal{J}} h_t(S) = 1 \quad \forall\, t\in\mathcal{T} \\
& h_t(S) \ge 0 \quad \forall\, S\subseteq\mathcal{J},\ t\in\mathcal{T}, \quad y_{jp}\ge 0 \quad \forall\, j\in\mathcal{J},\ p\in\mathcal{R}_j.
\end{aligned} \tag{8}
$$

Note that $\sum_{t\in\mathcal{T}} \sum_{S\subseteq\mathcal{J}} \phi_{jt}(S)\, h_t(S)$ is the total expected number of purchases for product j. The first constraint ensures that the total expected number of purchases for product j equals the total expected number of route assignments made for product j. By the second constraint, the expected number of route assignments that consume the capacity of resource i does not exceed the capacity of resource i. The third constraint implies that we offer an assortment with probability one at time period t. Linear programs similar to the linear program above have been used in the literature to obtain upper bounds on the optimal total expected revenue in numerous contexts. Accordingly, we can show that Z^*_{LP} is an upper bound on the optimal total expected revenue.

Since Z^*_{LP} is an upper bound on the optimal total expected revenue, it is enough to show that the total expected revenue of the approximate policy is at least $Z^*_{\mathsf{LP}}/(1+\theta L)$. The proof of Theorem 4.1 will have two steps. First, we show that we can use the parameters $(\hat{\gamma}_{jpt} : j\in\mathcal{J},\ p\in\mathcal{R}_j,\ t\in\mathcal{T})$

to upper bound on Z^*_{LP}. In particular, we show that $(1+\theta L)\sum_{j\in\mathcal{J}}\sum_{p\in\mathcal{R}_i}\hat{\gamma}_{jp1}\geq Z^*_{\mathsf{LP}}$. To show this result, we use the parameters $(\hat{\gamma}_{jpt}: j\in\mathcal{J},\ p\in\mathcal{R}_j,\ t\in\mathcal{T})$ to construct a feasible solution to the dual of problem (8) and this feasible dual solution provides an objective value of at least $(1+\theta L)\sum_{j\in\mathcal{J}}\sum_{p\in\mathcal{R}_i}\hat{\gamma}_{jp1}$. Second, we show that the total expected revenue of the approximate policy is at least $\hat{J}_1(\mathbf{0})$, where $\mathbf{0}\in\mathbb{R}_+^{|\mathcal{J}|}$ is the vector of all zeros. To show this result, letting $U_t(\boldsymbol{x})$ be the total expected revenue obtained by the approximate policy over time periods $t,\ldots,T$ starting with the state vector $\boldsymbol{x}$, we use induction over the time periods to show that $U_t(\boldsymbol{x})\geq\hat{J}_t(\boldsymbol{x})$. Lastly, if we solve problem (3) with $t=1$ and $\boldsymbol{x}=\mathbf{0}$, then the only feasible solution $\boldsymbol{w}$ to this problem has $w_i=c_i$ for all $i\in\mathcal{L}$, so letting $\boldsymbol{c}=(c_i: i\in\mathcal{L})$ and noting that $\psi_p(\boldsymbol{c})=1$ at this feasible solution, we get $\hat{J}_1(\mathbf{0})=\sum_{j\in\mathcal{J}}\sum_{p\in\mathcal{R}_j}\hat{\gamma}_{jp1}$. Putting the results together, the total expected revenue obtained by the approximate policy satisfies $U_1(\mathbf{0})\geq\hat{J}_1(\mathbf{0})=\sum_{j\in\mathcal{J}}\sum_{p\in\mathcal{R}_j}\hat{\gamma}_{jp1}\geq Z^*_{\mathsf{LP}}/(1+\theta L)$.

5. Performance Guarantee

In this section, we give a proof for Theorem 4.1. Associating the dual variables $(\alpha_j : j\in\mathcal{J})$, $(\mu_i : i\in\mathcal{L})$ and $(\sigma_t : t\in\mathcal{T})$ with the constraints, the dual of problem (8) is

$$
\begin{aligned}
\min\quad & \sum_{t\in\mathcal{T}}\sigma_t+\sum_{i\in\mathcal{L}}c_i\,\mu_i && (9)\\
\text{st}\quad & \sigma_t\geq\sum_{j\in\mathcal{J}}\phi_{jt}(S)\,(f_j-\alpha_j)\quad \forall\,S\subseteq\mathcal{J},\ t\in\mathcal{T}\\
& \sum_{i\in\mathcal{L}}a_{ip}\,\mu_i\geq\alpha_j\quad\forall\,j\in\mathcal{J},\ p\in\mathcal{R}_j\\
& \alpha_j,\ \sigma_t \text{ are free}\quad\forall\,j\in\mathcal{J},\ t\in\mathcal{T},\quad \mu_i\geq 0\quad\forall\,i\in\mathcal{L}.
\end{aligned}
$$

Problem (8) is feasible and bounded. In particular, setting $h_t(\varnothing)=1$ for all $t\in\mathcal{T}$ and the other decision variables to zero provides a feasible solution to this problem. The expected number of purchases for any product cannot exceed T, so the optimal objective value is bounded by $T\max_{j\in\mathcal{J}}f_j$. Therefore, the objective function of the dual above is also Z^*_{LP}. We make two observations to simplify problem (9). First, noting the objective function, we need to choose the value of σ_t as small as possible, in which case, by the first constraint, we have $\sigma_t=\max_{S\subseteq\mathcal{J}}\sum_{j\in\mathcal{J}}\phi_{jt}(S)\,(f_j-\alpha_j)$ in an optimal solution. Second, since we need to choose the value of σ_t as small as possible, by the first constraint, we need to choose the value of α_j as large as possible, so by the second constraint, we have $\alpha_j=\min_{p\in\mathcal{R}_j}\sum_{i\in\mathcal{L}}a_{ip}\,\mu_i$. Therefore, by our second observation, we have $f_j-\alpha_j=\max_{p\in\mathcal{R}_j}\{f_j-\sum_{i\in\mathcal{L}}a_{ip}\,\mu_i\}$, in which case, our first observation yields $\sigma_t=\max_{S\subseteq\mathcal{J}}\sum_{j\in\mathcal{J}}\phi_{jt}(S)\,\max_{p\in\mathcal{R}_j}\{f_j-\sum_{i\in\mathcal{L}}a_{ip}\,\mu_i\}$. Thus, replacing σ_t in the objective function of problem (9) with the last expression, using the vector of decision variables $\boldsymbol{\mu} = (\mu_i : i \in \mathcal{L})$, we can write problem (9) equivalently as

$$Z^*_{\mathsf{LP}} = \min_{\boldsymbol{\mu} \in \mathbb{R}^{|\mathcal{L}|}_+} \left\{ \sum_{t \in \mathcal{T}} \max_{S \subseteq \mathcal{J}} \left\{ \sum_{j \in \mathcal{J}} \phi_{jt}(S) \max_{p \in \mathcal{R}_j} \left\{ f_j - \sum_{i \in \mathcal{L}} a_{ip}\, \mu_i \right\} \right\} + \sum_{i \in \mathcal{L}} c_i\, \mu_i \right\}, \tag{10}$$

where the only decision variables are $(\mu_i : i \in \mathcal{L})$. In the next proposition, we use the substitutability property of the choice model to upper bound Z^*_{LP} with a function of $(\hat{\gamma}_{jp1} : j \in \mathcal{J},\ p \in \mathcal{R}_j)$.

Proposition 5.1 (Optimal Performance Benchmark) *Letting* $(\hat{\gamma}_{jpt} : j \in \mathcal{J},\ p \in \mathcal{R}_j,\ t \in \mathcal{T})$ *be computed through* (4)-(6), *we have* $(1 + \theta L) \sum_{j \in \mathcal{J}} \sum_{p \in \mathcal{R}_j} \hat{\gamma}_{jp1} \geq Z^*_{\mathsf{LP}}$.

Proof: Let $\Delta_{jt} = f_j - \theta \sum_{i \in A_{\hat{p}_{jt}}} \frac{1}{c_i} \sum_{k \in \mathcal{J}} \sum_{q \in \mathcal{R}_k} a_{iq}\, \hat{\gamma}_{kq,t+1}$ for notational brevity, so problem (5) is of the form $\hat{S}_t = \arg\max_{S \subseteq \mathcal{J}} \sum_{j \in \mathcal{J}} \phi_{jt}(S)\, \Delta_{jt}$. We can use a reasoning similar to the one right after (2) to argue that if $j \in \hat{S}_t$ in (5), then $\Delta_{jt} \geq 0$. In particular, assume that $\Delta_{jt} < 0$ for some $j \in \hat{S}_t$. If we drop each product k with $\Delta_{kt} < 0$ from $\hat{S}_t$, then the substitutability property of the choice model implies that the choice probability of all other products in $\hat{S}_t$ increases. In this way, we eliminate each product k with $\Delta_{kt} < 0$ from $\hat{S}_t$ and the remaining products have even larger choice probabilities, so the solution that we obtain in this way must also be an optimal solution to problem (5). By (6), if $p = \hat{p}_{jt}$, then $\hat{\gamma}_{jpt} = \phi_{jt}(\hat{S}_t)\, \Delta_{jt} + \hat{\gamma}_{jp,t+1}$, whereas if $p \neq \hat{p}_{jt}$, then $\hat{\gamma}_{jpt} = \hat{\gamma}_{jp,t+1}$. Thus, because $\Delta_{jt} \geq 0$ for all $j \in \hat{S}_t$, we get $\hat{\gamma}_{jpt} \geq \hat{\gamma}_{jp,t+1}$. By the boundary condition $\hat{\gamma}_{jp,T+1} = 0$, we get $\hat{\gamma}_{jp1} \geq \hat{\gamma}_{jp2} \geq \ldots \geq \hat{\gamma}_{jpT} \geq \hat{\gamma}_{jp,T+1} = 0$. We define a solution to problem (10) as $\hat{\mu}_i = \frac{\theta}{c_i} \sum_{k \in \mathcal{J}} \sum_{q \in \mathcal{R}_j} a_{iq}\, \hat{\gamma}_{kq1}$ for all $i \in \mathcal{L}$. Since $\hat{\gamma}_{jp1} \geq 0$, this solution is feasible to (10). Evaluating the objective value of problem (10) at this feasible solution, we upper bound Z^*_{LP}, so

$$\begin{aligned}
Z^*_{\mathsf{LP}} \;&\leq\; \sum_{t \in \mathcal{T}} \max_{S \subseteq \mathcal{J}} \left\{ \sum_{j \in \mathcal{J}} \phi_{jt}(S) \max_{p \in \mathcal{R}_j} \left\{ f_j - \sum_{i \in A_p} \hat{\mu}_i \right\} \right\} + \sum_{i \in \mathcal{L}} c_i\, \hat{\mu}_i \\
&\overset{(a)}{=} \sum_{t \in \mathcal{T}} \max_{S \subseteq \mathcal{J}} \left\{ \sum_{j \in \mathcal{J}} \phi_{jt}(S) \max_{p \in \mathcal{R}_j} \left\{ f_j - \theta \sum_{i \in A_p} \frac{1}{c_i} \sum_{k \in \mathcal{J}} \sum_{q \in \mathcal{R}_j} a_{iq}\, \hat{\gamma}_{kq1} \right\} \right\} + \theta \sum_{i \in \mathcal{L}} \sum_{k \in \mathcal{J}} \sum_{q \in \mathcal{R}_j} a_{iq}\, \hat{\gamma}_{kq1} \\
&\overset{(b)}{\leq} \sum_{t \in \mathcal{T}} \max_{S \subseteq \mathcal{J}} \left\{ \sum_{j \in \mathcal{J}} \phi_{jt}(S) \max_{p \in \mathcal{R}_j} \left\{ f_j - \theta \sum_{i \in A_p} \frac{1}{c_i} \sum_{k \in \mathcal{J}} \sum_{q \in \mathcal{R}_j} a_{iq}\, \hat{\gamma}_{kq,t+1} \right\} \right\} + \theta L \sum_{k \in \mathcal{J}} \sum_{q \in \mathcal{R}_j} \hat{\gamma}_{kq1} \\
&\overset{(c)}{=} \sum_{t \in \mathcal{T}} \max_{S \subseteq \mathcal{J}} \left\{ \sum_{j \in \mathcal{J}} \phi_{jt}(S) \left(f_j - \theta \sum_{i \in A_{\hat{p}_{jt}}} \frac{1}{c_i} \sum_{k \in \mathcal{J}} \sum_{q \in \mathcal{R}_j} a_{iq}\, \hat{\gamma}_{kq,t+1} \right) \right\} + \theta L \sum_{k \in \mathcal{J}} \sum_{q \in \mathcal{R}_j} \hat{\gamma}_{kq1} \\
&\overset{(d)}{=} \sum_{t \in \mathcal{T}} \sum_{j \in \mathcal{J}} \phi_{jt}(\hat{S}_t) \left(f_j - \theta \sum_{i \in A_{\hat{p}_{jt}}} \frac{1}{c_i} \sum_{k \in \mathcal{J}} \sum_{q \in \mathcal{R}_j} a_{iq}\, \hat{\gamma}_{kq,t+1} \right) + \theta L \sum_{k \in \mathcal{J}} \sum_{q \in \mathcal{R}_j} \hat{\gamma}_{kq1}
\end{aligned}$$

In the chain of inequalities above, (a) is by the definition of $\hat{\mu}_i$, (b) holds because $\hat{\gamma}_{kp1} \geq \hat{\gamma}_{kp,t+1}$ by the discussion in the previous paragraph and $L \geq |A_q| = \sum_{i \in \mathcal{L}} a_{iq}$ for any $j \in \mathcal{J}$ and $q \in \mathcal{R}_j$, (c)

holds by the definition of $\hat{p}_{jt}$ in (4) and (d) holds by the definition of $\hat{S}_t$ in (5). Observe that we can express the right side of the chain of inequalities above equivalently as

$$\sum_{t\in\mathcal{T}}\sum_{j\in\mathcal{J}}\sum_{p\in\mathcal{R}_j}\phi_{jt}(\hat{S}_t)\,\mathbf{1}_{(p=\hat{p}_{jt})}\left(f_j-\theta\sum_{i\in A_p}\frac{1}{c_i}\sum_{k\in\mathcal{J}}\sum_{q\in\mathcal{R}_j}a_{iq}\,\hat{\gamma}_{kq,t+1}\right)+\theta L\sum_{k\in\mathcal{J}}\sum_{q\in\mathcal{R}_j}\hat{\gamma}_{kq1}$$
$$\stackrel{(e)}{=}\sum_{t\in\mathcal{T}}\sum_{j\in\mathcal{J}}\sum_{p\in\mathcal{R}_j}(\hat{\gamma}_{jpt}-\hat{\gamma}_{jp,t+1})+\theta L\sum_{k\in\mathcal{J}}\sum_{q\in\mathcal{R}_j}\hat{\gamma}_{kq1}\stackrel{(f)}{=}(1+\theta L)\sum_{j\in\mathcal{J}}\sum_{p\in\mathcal{R}_j}\hat{\gamma}_{jp1},$$

where (e) uses (6) and (f) follows by canceling the telescoping terms in the first sum. Collecting the two chains of inequalities above yields the desired result. ■

Proposition 5.1 is the first step for the proof of Theorem 4.1 discussed in the previous section. The second step uses two lemmas. In the next lemma, we upper bound the opportunity cost.

Lemma 5.2 (Opportunity Cost) *Recalling that $\hat{\Psi}_t(\boldsymbol{w})=\sum_{j\in\mathcal{J}}\sum_{p\in\mathcal{R}_j}\hat{\gamma}_{jpt}\,\psi_p(\boldsymbol{w})$ and $\overline{\boldsymbol{e}}_i\in\mathbb{Z}_+^{|\mathcal{L}|}$ is the i-th unit vector, for any $\boldsymbol{w}\in\mathbb{Z}_+^{|\mathcal{L}|}$, we have*

$$\hat{\Psi}_t(\boldsymbol{w})-\hat{\Psi}_t\left(\boldsymbol{w}-\sum_{i\in A_p}\overline{\boldsymbol{e}}_i\right)\;\le\;\sum_{i\in A_p}\frac{1}{c_i}\sum_{k\in\mathcal{J}}\sum_{q\in\mathcal{R}_k}a_{iq}\,\hat{\gamma}_{kqt}.$$

Proof: For two vectors $\boldsymbol{\alpha}=(\alpha_1,\ldots,\alpha_n)$ and $\boldsymbol{\beta}=(\beta_1,\ldots,\beta_n)$, we have the standard inequality $\min_i\alpha_i-\min_i\beta_i\le\sum_{i=1}^n|\alpha_i-\beta_i|$. Note that the i-th component of the vector $\boldsymbol{w}-\sum_{i\in A_p}\overline{\boldsymbol{e}}_i$ is w_i-a_{ip}. Therefore, using the definition of $\psi_p(\boldsymbol{w})$, we obtain $\psi_q(\boldsymbol{w})-\psi_q(\boldsymbol{w}-\sum_{i\in A_p}\overline{\boldsymbol{e}}_i)=\min_{i\in A_q}\{\frac{w_i}{c_i}\}-\min_{i\in A_q}\{\frac{w_i-a_{ip}}{c_i}\}\le\sum_{i\in A_q}\frac{a_{ip}}{c_i}$, which yields

$$\hat{\Psi}_t(\boldsymbol{w})-\hat{\Psi}_t\left(\boldsymbol{w}-\sum_{i\in A_p}\overline{\boldsymbol{e}}_i\right)\;=\;\sum_{k\in\mathcal{J}}\sum_{q\in\mathcal{R}_k}\hat{\gamma}_{kqt}\,\min_{i\in A_q}\left\{\frac{w_i}{c_i}\right\}-\sum_{k\in\mathcal{J}}\sum_{q\in\mathcal{R}_k}\hat{\gamma}_{kqt}\,\min_{i\in A_q}\left\{\frac{w_i-a_{ip}}{c_i}\right\}$$
$$\le\;\sum_{k\in\mathcal{J}}\sum_{q\in\mathcal{R}_k}\sum_{i\in A_q}\hat{\gamma}_{kqt}\,\frac{a_{ip}}{c_i}\stackrel{(a)}{=}\sum_{k\in\mathcal{J}}\sum_{q\in\mathcal{R}_k}\sum_{i\in\mathcal{L}}a_{iq}\,\hat{\gamma}_{kqt}\,\frac{a_{ip}}{c_i}\stackrel{(b)}{=}\sum_{i\in A_p}\frac{1}{c_i}\sum_{k\in\mathcal{J}}\sum_{q\in\mathcal{R}_k}a_{iq}\,\hat{\gamma}_{kqt},$$

where (a) holds because $i\in A_q$ if and only if $a_{iq}=1$ and (b) follows by arranging the terms and using the fact that $a_{ip}=1$ if and inly if $i\in A_p$. ■

In the next lemma, we use a feasibility argument in problem (3) to lower bound the value function approximation after the purchase for a product.

Lemma 5.3 (Value Function Approximation Bound) *For any $\boldsymbol{x}\in\mathbb{Z}_+^{|\mathcal{J}|}$, $\boldsymbol{y}\in\mathcal{F}(\boldsymbol{x})$, $j\in\mathcal{J}$ and $p\in\mathcal{R}_j$, letting $w_i=c_i-\sum_{k\in\mathcal{J}}\sum_{q\in\mathcal{R}_k}a_{iq}\,y_{kq}$ for all $i\in\mathcal{L}$, if we have $w_i\ge 1$ for all $i\in A_p$, then $\mathcal{F}(\boldsymbol{x}+\boldsymbol{e}_j)\neq\varnothing$ and $\hat{J}_t(\boldsymbol{x}+\boldsymbol{e}_j)\ge\hat{\Psi}_t(\boldsymbol{w}-\sum_{i\in A_p}\overline{\boldsymbol{e}}_i)$.*

Proof: Fixing $\boldsymbol{x}\in\mathbb{Z}_+^{|\mathcal{J}|}$, $\boldsymbol{y}=\mathcal{F}(\boldsymbol{x})$, $j\in\mathcal{J}$ and $p\in\mathcal{R}_j$, let $\boldsymbol{w}=(w_i:i\in\mathcal{L})$ be as given in the lemma. Define $\widehat{\boldsymbol{y}}=(\hat{y}_{kq}:k\in\mathcal{J},\ q\in\mathcal{R}_k)$ as $\hat{y}_{kq}=y_{kq}+\mathbf{1}_{((k,q)=(j,p))}$. We have $\sum_{q\in\mathcal{R}_k}\hat{y}_{kq}=\sum_{q\in\mathcal{R}_k}y_{kq}+$

$\sum_{q \in \mathcal{R}_k} \mathbf{1}_{((k,q)=(j,p))} = \sum_{q \in \mathcal{R}_k} y_{kq} + \mathbf{1}_{(k=j)} = x_k + \mathbf{1}_{(k=j)}$, where the last equality holds since $\boldsymbol{y} \in \mathcal{F}(\boldsymbol{x})$. Thus, $\widehat{\boldsymbol{y}}$ satisfies the first constraint when we replace $\boldsymbol{x}$ in (1) with $\boldsymbol{x} + \boldsymbol{e}_j$. Also, we have

$$\sum_{k \in \mathcal{J}} \sum_{q \in \mathcal{R}_k} a_{iq}\, \hat{y}_{kq} = \sum_{k \in \mathcal{J}} \sum_{q \in \mathcal{R}_k} a_{iq}\, y_{kq} + \sum_{k \in \mathcal{J}} \sum_{q \in \mathcal{R}_k} a_{iq}\, \mathbf{1}_{((k,q)=(j,p))} = \sum_{k \in \mathcal{J}} \sum_{q \in \mathcal{R}_k} a_{iq}\, y_{kq} + a_{ip} \overset{(a)}{=} c_i - w_i + a_{ip} \overset{(b)}{\leq} c_i,$$

where (a) uses the definition of w_i and (b) uses the fact that $w_i \geq 1$ for all $i \in A_p$, which is equivalent to $w_i \geq a_{ip}$ for all $i \in \mathcal{L}$. Thus, $\widehat{\boldsymbol{y}}$ satisfies the second constraint when we replace $\boldsymbol{x}$ in (1) with $\boldsymbol{x} + \boldsymbol{e}_j$. In this case, we get $\widehat{\boldsymbol{y}} \in \mathcal{F}(\boldsymbol{x} + \boldsymbol{e}_j)$, so $\mathcal{F}(\boldsymbol{x} + \boldsymbol{e}_j) \neq \varnothing$, establishing the first statement in the lemma. By the first and fourth expressions in the chain of equalities above, we have $\sum_{k \in \mathcal{J}} \sum_{q \in \mathcal{R}_k} a_{iq}\, \hat{y}_{kq} + w_i - a_{ip} = c_i$. Thus, noting that the i-th component of the vector $\boldsymbol{w} - \sum_{i \in A_p} \overline{\boldsymbol{e}}_i$ is $w_i - a_{ip}$, the last equality shows that $(\widehat{\boldsymbol{y}}, \boldsymbol{w} - \sum_{i \in A_p} \overline{\boldsymbol{e}}_i)$ satisfies the second constraint in problem (3) when we replace $\boldsymbol{x}$ with $\boldsymbol{x} + \boldsymbol{e}_j$. By the discussion at the beginning of the proof, $(\widehat{\boldsymbol{y}}, \boldsymbol{w} - \sum_{i \in A_p} \overline{\boldsymbol{e}}_i)$ also satisfies the first constraint in problem (3) when we replace $\boldsymbol{x}$ with $\boldsymbol{x} + \boldsymbol{e}_j$. Since the optimal objective value of this problem is $\hat{J}_t(\boldsymbol{x} + \boldsymbol{e}_j)$, we get $\hat{J}_t(\boldsymbol{x} + \boldsymbol{e}_j) \geq \hat{\Psi}_t(\boldsymbol{w} - \sum_{i \in A_p} \overline{\boldsymbol{e}}_i)$, which is the second statement in the lemma. ■

Let $U_t(\boldsymbol{x})$ be the total expected revenue obtained by the approximate policy over time periods $t, \ldots, T$ starting with the state vector $\boldsymbol{x}$ at time period t. We can compute $\{U_t : t \in \mathcal{T}\}$ by

$$U_t(\boldsymbol{x}) = \sum_{j \in \mathcal{J}} \phi_{jt}(S_t^{\mathsf{App}}(\boldsymbol{x}))\, \mathbf{1}_{(\mathcal{F}(\boldsymbol{x}+\boldsymbol{e}_j) \neq \varnothing)} \Big[f_j + U_{t+1}(\boldsymbol{x} + \boldsymbol{e}_j) - U_{t+1}(\boldsymbol{x}) \Big] + U_{t+1}(\boldsymbol{x}), \tag{11}$$

with the boundary condition $U_{T+1}(\boldsymbol{x}) = 0$ for all $\boldsymbol{x} \in \mathbb{Z}_+^{|\mathcal{J}|}$. The dynamic program above is similar to the one in (2), but the decision at each time period in (11) is fixed by the approximate policy, as given in (7). We refer to $\{U_t : t \in \mathcal{T}\}$ as the value functions of the approximate policy. One useful observation is that if we arrange the terms on the right side of (11), then the coefficient of $U_{t+1}(\boldsymbol{x})$ is $1 - \sum_{j \in \mathcal{J}} \phi_{jt}(S_t^{\mathsf{App}}(\boldsymbol{x}))\, \mathbf{1}_{(\mathcal{F}(\boldsymbol{x}+\boldsymbol{e}_j) \neq \varnothing)}$. Noting that $\sum_{j \in \mathcal{J}} \phi_{jt}(S_t^{\mathsf{App}}(\boldsymbol{x})) \leq 1$, this coefficient is non-negative. Thus, if we replace $U_{t+1}(\boldsymbol{x})$ on the right side of (11) with a larger quantity, then the right side of (11) becomes larger. This observation will shortly become useful. In the next proposition, using the two lemmas above, we focus on the second part of the proof of Theorem 4.1 discussed in the previous section. In particular, we show that the value functions of the approximate policy are lower bounded by our value function approximations.

Proposition 5.4 (Approximate Policy Performance Benchmark) *For any $\boldsymbol{x} \in \mathbb{Z}_+^{|\mathcal{J}|}$ with $\mathcal{F}(\boldsymbol{x}) \neq \varnothing$ and $t \in \mathcal{T}$, we have $U_t(\boldsymbol{x}) \geq \hat{J}_t(\boldsymbol{x})$.*

Proof: We show the result by induction over the time periods. We have $U_{T+1}(\boldsymbol{x}) = 0 = \hat{J}_{T+1}(\boldsymbol{x})$, so the result holds at time period $T+1$. Assuming that the result holds at time period $t+1$, we show that the result holds at time period t. Throughout the proof, we fix the state vector $\boldsymbol{x}$. Let $(\widehat{\boldsymbol{y}}, \widehat{\boldsymbol{w}})$ be an optimal solution to problem (3) with the value of the state vector we fix. Therefore, we have $\widehat{\boldsymbol{y}} \in \mathcal{F}(\boldsymbol{x})$, $\hat{w}_i = c_i - \sum_{j\in\mathcal{J}} \sum_{p\in\mathcal{R}_k} a_{ip}\, \hat{y}_{jp}$ for all $i \in \mathcal{L}$ and $\hat{J}_t(\boldsymbol{x}) = \hat{\Psi}_t(\widehat{\boldsymbol{w}})$. We make three observations. First, $(\widehat{\boldsymbol{y}}, \widehat{\boldsymbol{w}})$ is a feasible solution to problem (3) when we solve this problem for the value of the state vector we fix but for time period $t+1$. Thus, $\hat{J}_{t+1}(\boldsymbol{x}) \geq \hat{\Psi}_{t+1}(\widehat{\boldsymbol{w}})$. Second, since $\widehat{\boldsymbol{y}} \in \mathcal{F}(\boldsymbol{x})$, by Lemma 5.3, for any $j \in \mathcal{J}$ and $p \in \mathcal{R}_j$, having $\hat{w}_i \geq 1$ for all $i \in A_p$ implies that $\mathcal{F}(\boldsymbol{x}+\boldsymbol{e}_j) \neq \varnothing$ and $\hat{J}_t(\boldsymbol{x}+\boldsymbol{e}_j) \geq \hat{\Psi}_t(\widehat{\boldsymbol{w}} - \sum_{i\in A_p} \overline{\boldsymbol{e}}_i)$. Thus, for any $j \in \mathcal{J}$ and $p \in \mathcal{R}_j$, we have

$$\prod_{i\in A_p} \mathbf{1}_{(\hat{w}_i \geq 1)} \leq \mathbf{1}_{(\mathcal{F}(\boldsymbol{x}+\boldsymbol{e}_j)\neq\varnothing)}, \tag{12}$$

$$\Big(\prod_{i\in A_p} \mathbf{1}_{(\hat{w}_i \geq 1)}\Big) \hat{J}_t(\boldsymbol{x}+\boldsymbol{e}_j) \geq \Big(\prod_{i\in A_p} \mathbf{1}_{(\hat{w}_i \geq 1)}\Big) \hat{\Psi}_t\Big(\widehat{\boldsymbol{w}} - \sum_{i\in A_p} \overline{\boldsymbol{e}}_i\Big). \tag{13}$$

Third, by the same reasoning right after (2), there exists an optimal solution to problem (7) such that if we have $f_j + \hat{J}_{t+1}(\boldsymbol{x}+\boldsymbol{e}_j) - \hat{J}_{t+1}(\boldsymbol{x}) \leq 0$ in (7), then $j \notin S_t^{\mathsf{App}}(\boldsymbol{x})$.

By the induction argument, $U_{t+1}(\boldsymbol{x}) \geq \hat{J}_{t+1}(\boldsymbol{x})$ and $U_{t+1}(\boldsymbol{x}+\boldsymbol{e}_j) \geq \hat{J}_{t+1}(\boldsymbol{x}+\boldsymbol{e}_j)$. By the discussion right before the theorem, the right side of (11) is increasing in $U_{t+1}(\boldsymbol{x})$, so (11) implies

$$\begin{aligned} U_t(\boldsymbol{x}) \;\geq\; & \sum_{j\in\mathcal{J}} \phi_{jt}(S_t^{\mathsf{App}}(\boldsymbol{x}))\, \mathbf{1}_{(\mathcal{F}(\boldsymbol{x}+\boldsymbol{e}_j)\neq\varnothing)} \Big[f_j + \hat{J}_{t+1}(\boldsymbol{x}+\boldsymbol{e}_j) - \hat{J}_{t+1}(\boldsymbol{x})\Big] + \hat{J}_{t+1}(\boldsymbol{x}) \\ \stackrel{(a)}{=}\; & \sum_{j\in\mathcal{J}} \phi_{jt}(S_t^{\mathsf{App}}(\boldsymbol{x}))\, \mathbf{1}_{(\mathcal{F}(\boldsymbol{x}+\boldsymbol{e}_j)\neq\varnothing)} \Big[f_j + \hat{J}_{t+1}(\boldsymbol{x}+\boldsymbol{e}_j) - \hat{J}_{t+1}(\boldsymbol{x})\Big]^+ + \hat{J}_{t+1}(\boldsymbol{x}) \\ \stackrel{(b)}{\geq}\; & \sum_{j\in\mathcal{J}} \phi_{jt}(\hat{S}_t)\, \mathbf{1}_{(\mathcal{F}(\boldsymbol{x}+\boldsymbol{e}_j)\neq\varnothing)} \Big[f_j + \hat{J}_{t+1}(\boldsymbol{x}+\boldsymbol{e}_j) - \hat{J}_{t+1}(\boldsymbol{x})\Big]^+ + \hat{J}_{t+1}(\boldsymbol{x}), \end{aligned} \tag{14}$$

where (a) holds because we can assume that $f_j + \hat{J}_{t+1}(\boldsymbol{x}+\boldsymbol{e}_j) - \hat{J}_{t+1}(\boldsymbol{x}) > 0$ for all $j \in S_t^{\mathsf{App}}(\boldsymbol{x})$ by the third observation and (b) holds because $\hat{S}_t$ in (5) may not be optimal to problem (7).

All of the terms in the last sum in (14) are non-negative, in which case, using (12) with the ideal route $\hat{p}_{jt}$ for product j in (4), we can lower bound the right side of (14) as

$$\begin{aligned} & \sum_{j\in\mathcal{J}} \phi_{jt}(\hat{S}_t) \prod_{i\in A_{\hat{p}_{jt}}} \mathbf{1}_{(\hat{w}_i \geq 1)} \Big[f_j + \hat{J}_{t+1}(\boldsymbol{x}+\boldsymbol{e}_j) - \hat{J}_{t+1}(\boldsymbol{x})\Big]^+ + \hat{J}_{t+1}(\boldsymbol{x}) \\ & \stackrel{(c)}{\geq} \sum_{j\in\mathcal{J}} \phi_{jt}(\hat{S}_t) \prod_{i\in A_{\hat{p}_{jt}}} \mathbf{1}_{(\hat{w}_i \geq 1)} \Big[f_j + \hat{\Psi}_{t+1}\Big(\widehat{\boldsymbol{w}} - \sum_{i\in A_{\hat{p}_{jt}}} \overline{\boldsymbol{e}}_i\Big) - \hat{J}_{t+1}(\boldsymbol{x})\Big]^+ + \hat{J}_{t+1}(\boldsymbol{x}) \\ & \stackrel{(d)}{\geq} \sum_{j\in\mathcal{J}} \phi_{jt}(\hat{S}_t) \prod_{i\in A_{\hat{p}_{jt}}} \mathbf{1}_{(\hat{w}_i \geq 1)} \Big[f_j + \hat{\Psi}_{t+1}\Big(\widehat{\boldsymbol{w}} - \sum_{i\in A_{\hat{p}_{jt}}} \overline{\boldsymbol{e}}_i\Big) - \hat{\Psi}_{t+1}(\widehat{\boldsymbol{w}})\Big]^+ + \hat{\Psi}_{t+1}(\widehat{\boldsymbol{w}}), \end{aligned} \tag{15}$$

where (c) follows by (13), whereas (d) uses the fact that $\hat{J}_{t+1}(\boldsymbol{x}) \geq \hat{\Psi}_{t+1}(\widehat{\boldsymbol{w}})$ by the first observation, as well as noting that $\sum_{i=1}^n \delta_i [x-y]^+ + y$ is increasing in y when $\delta_i \geq 0$ for all $i = 1, \ldots, n$ and

$\sum_{i=1}^n \delta_i \leq 1$. We can use Lemma 5.2 to upper bound the difference $\hat{\Psi}_{t+1}(\widehat{\boldsymbol{w}}) - \hat{\Psi}_{t+1}(\widehat{\boldsymbol{w}} - \sum_{i \in A_{\hat{p}_{jt}}} \overline{\boldsymbol{e}}_i)$, in which case, we can lower bound the right side of (15) as

$$\sum_{j\in\mathcal{J}} \phi_{jt}(\hat{S}_t) \prod_{i \in A_{\hat{p}_{jt}}} \mathbf{1}_{(\hat{w}_i \geq 1)} \Big[f_j - \sum_{i\in A_{\hat{p}_{jt}}} \frac{1}{c_i} \sum_{k\in\mathcal{J}} \sum_{q \in \mathcal{R}_k} a_{iq}\, \hat{\gamma}_{kq,t+1} \Big]^+ + \hat{\Psi}_{t+1}(\widehat{\boldsymbol{w}})$$

$$= \sum_{j\in\mathcal{J}} \sum_{p\in\mathcal{R}_j} \phi_{jt}(\hat{S}_t)\, \mathbf{1}_{(p=\hat{p}_{jt})} \prod_{i\in A_p} \mathbf{1}_{(\hat{w}_i\geq 1)} \Big[f_j - \sum_{i\in A_p} \frac{1}{c_i} \sum_{k\in\mathcal{J}} \sum_{q\in\mathcal{R}_k} a_{iq}\, \hat{\gamma}_{kq,t+1} \Big]^+ + \hat{\Psi}_{t+1}(\widehat{\boldsymbol{w}})$$

$$\stackrel{(e)}{\geq} \sum_{j\in\mathcal{J}} \sum_{p\in\mathcal{R}_j} \phi_{jt}(\hat{S}_t)\, \mathbf{1}_{(p=\hat{p}_{jt})}\, \psi_p(\widehat{\boldsymbol{w}}) \Big[f_j - \sum_{i\in A_p} \frac{1}{c_i} \sum_{k\in\mathcal{J}} \sum_{q\in\mathcal{R}_k} a_{iq}\, \hat{\gamma}_{kq,t+1} \Big]^+ + \hat{\Psi}_{t+1}(\widehat{\boldsymbol{w}}), \quad (16)$$

where (e) holds by noting that $\mathbf{1}_{(z\geq 1)} \geq \frac{z}{c_i}$ for any $z \in \mathbb{Z}_+$ with $0 \leq z \leq c_i$, in which case, the definition of ψ_p implies that $\prod_{i\in A_p} \mathbf{1}_{(\hat{w}_i \geq 1)} \geq \min_{i\in A_p}\{\frac{\hat{w}_i}{c_i}\} = \psi_p(\widehat{\boldsymbol{w}})$.

If we do not round the term in the square brackets on the right side of (16) up to zero, then this term becomes smaller. Also, noting that $\theta \geq 1$, we lower bound the right side of (16) as

$$\sum_{j\in\mathcal{J}} \sum_{p\in\mathcal{R}_j} \phi_{jt}(\hat{S}_t)\, \mathbf{1}_{(p=\hat{p}_{jt})}\, \psi_p(\widehat{\boldsymbol{w}}) \Big[f_j - \theta \sum_{i\in A_p} \frac{1}{c_i} \sum_{k\in\mathcal{J}} \sum_{q\in\mathcal{R}_k} a_{iq}\, \hat{\gamma}_{kq,t+1} \Big] + \hat{\Psi}_{t+1}(\widehat{\boldsymbol{w}})$$

$$\stackrel{(f)}{=} \sum_{j\in\mathcal{J}} \sum_{p\in\mathcal{R}_j} (\hat{\gamma}_{jpt} - \hat{\gamma}_{jp,t+1})\, \psi_p(\widehat{\boldsymbol{w}}) + \hat{\Psi}_{t+1}(\widehat{\boldsymbol{w}}) \stackrel{(g)}{=} \hat{\Psi}_t(\widehat{\boldsymbol{w}}) \stackrel{(h)}{=} \hat{J}_t(\boldsymbol{x}), \quad (17)$$

where (f) holds by (6), (g) is by the definition of $\hat{\Psi}_t$ and (h) follows by the definition of $\widehat{\boldsymbol{w}}$. Collecting (14)-(17), we have $U_t(\boldsymbol{x}) \geq \hat{J}_t(\boldsymbol{x})$, which completes the induction argument. ■

Below, we use Propositions 5.1 and 5.4 to give a proof for Theorem 4.1.

Proof of Theorem 4.1: Noting that we do not have any purchases for any products at the beginning of the selling horizon, the total expected revenue of the approximate policy is $U_1(\mathbf{0})$, whereas the optimal total expected revenue is $J_1(\mathbf{0})$. In Appendix B, we show that the optimal objective value of the linear program in (8) is an upper bound on the optimal total expected revenue, so $Z^*_{\mathsf{LP}} \geq J_1(\mathbf{0})$. This result follows by using the decisions of the optimal policy to construct a feasible solution to problem (8). On the other hand, if we solve problem (3) with $t = 1$ and $\boldsymbol{x} = \mathbf{0}$, then the only feasible solution to this problem must have $w_i = c_i$ for all $i \in \mathcal{L}$. Thus, using the vector $\boldsymbol{c} = (c_i : i \in \mathcal{L})$, since $\psi_p(\boldsymbol{c}) = 1$, we get $\hat{J}_1(\mathbf{0}) = \sum_{j\in\mathcal{J}} \sum_{p\in\mathcal{R}_j} \hat{\gamma}_{jp1}\, \psi_p(\boldsymbol{c}) = \sum_{j\in\mathcal{J}} \sum_{p\in\mathcal{R}_j} \hat{\gamma}_{jp1}$. In this case, by Propositions 5.1 and 5.4, we get $U_1(\mathbf{0}) \geq \hat{J}_1(\mathbf{0}) = \sum_{j\in\mathcal{J}} \sum_{p\in\mathcal{R}_j} \hat{\gamma}_{jp1} \geq \frac{1}{1+\theta L} Z^*_{\mathsf{LP}} \geq \frac{1}{1+\theta L} J_1(\mathbf{0})$.

6. Periodic Route Assignments

By making the route assignments at the end of the selling horizon, we pool the purchases over the whole selling horizon without committing to a route assignment until the end. While this approach allows us to make use of the resource capacities most efficiently, customers do not know what routes they are assigned to until the end. A sensible approach to strike a tradeoff between making use of the resource capacities most efficiently and letting the customers know what routes they are assigned to in a timely manner is to designate a set of time periods as route assignment periods. At each of these route assignment periods, we make irrevocable route assignments for the product purchases that have occurred since the last route assignment period. In this way, the customers do not have to wait until the end of the selling horizon to know what routes they are assigned to, but we also do not have to make a route assignment right after each purchase. We can extend our approximate policy to the case where we make the route assignments periodically, while still maintaining the performance guarantee of $1/(1+L)$. In this section, we discuss the main points of this extension, deferring the detailed analysis to Appendix C. We use this extension in our numerical experiments to study the revenue implications of making irrevocable route assignments periodically, instead of delaying the route assignments to the end of the selling horizon.

Dynamic Programming Formulation:

We use the same notation in Section 2, adding two pieces. We use $\mathcal{A} \subseteq \mathcal{T}$ to denote the set of route assignment periods. If $\mathcal{A} = \{T\}$, then we delay the route assignments until the end of the selling horizon. If, on the other hand, $\mathcal{A} = \{1, \ldots, T\}$, then we make a route assignment for each product purchase immediately. The set of route assignment periods can be anywhere between these two extremes, but it is fixed a priori. Let $\mathcal{R} = \cup_{j \in \mathcal{J}} \mathcal{R}_j$ be the set of all routes. The state of the system at the beginning of a generic time period has two components. Letting x_j be the number of customers with a purchase for product j since the last route assignment period, the first component of the state is $\boldsymbol{x} = (x_j : j \in \mathcal{J}) \in \mathbb{Z}_+^{|\mathcal{J}|}$. Letting z_p be the number of purchases that have been irrevocably assigned to route p, the second component of the state is $\boldsymbol{z} = (z_p : p \in \mathcal{R}) \in \mathbb{Z}_+^{|\mathcal{R}|}$. Therefore, we use $(\boldsymbol{x}, \boldsymbol{z}) \in \mathbb{Z}_+^{|\mathcal{J}|+|\mathcal{R}|}$ to represent the state of the system at the beginning of a generic time period. Let y_{jp} be the number of purchases for product j that we assign to route p. Using the decision variables $\boldsymbol{y} = (y_{jp} : j \in \mathcal{J},\ p \in \mathcal{R}_j) \in \mathbb{Z}_+^{\sum_{j \in \mathcal{J}} |\mathcal{R}_j|}$, if the state of the system at the beginning of a time period is $(\boldsymbol{x}, \boldsymbol{z})$, then the set of feasible route assignments is given by

$$\mathcal{F}(\boldsymbol{x}, \boldsymbol{z}) = \left\{ \boldsymbol{y} \in \mathbb{Z}_+^{\sum_{j \in \mathcal{J}} |\mathcal{R}_j|} : \sum_{p \in \mathcal{R}_j} y_{jp} = x_j \quad \forall j \in \mathcal{J}, \quad \sum_{j \in \mathcal{J}} \sum_{p \in \mathcal{R}_j} a_{ip}\, y_{jp} + \sum_{p \in \mathcal{R}} a_{ip}\, z_p \le c_i \quad \forall i \in \mathcal{L} \right\}, \tag{18}$$

which is similar to (1), but the feasible set above considers the fact that we cannot make route assignments for the purchases that have already been irrevocably assigned to a route.

If time period t is a route assignment period, then we make the route assignments after observing the product purchase, if any, at time period t. Given that the state of the system is $(\boldsymbol{x}, \boldsymbol{z})$ after observing the product purchase at time period t, we use $\mathcal{G}_t(\boldsymbol{x},\boldsymbol{z})$ to denote the set of possible states at the beginning of time period $t+1$. If $t \notin \mathcal{A}$, so that time period t is not a route assignment period, then the state of the system cannot change after observing the product purchase at time period t. Thus, we have $\mathcal{G}_t(\boldsymbol{x},\boldsymbol{z}) = \{(\boldsymbol{x},\boldsymbol{z})\}$. If $t \in \mathcal{A}$, so that time period t is a route assignment period, then we need to make route assignments for all product purchases without route assignments. Thus, we have $\mathcal{G}_t(\boldsymbol{x},\boldsymbol{z}) = \{(\boldsymbol{0},\overline{\boldsymbol{z}}) : \exists\, \boldsymbol{y} \in \mathcal{F}(\boldsymbol{x},\boldsymbol{z}) : \overline{z}_p = z_p + \sum_{j\in\mathcal{J}} \mathbf{1}_{(p\in\mathcal{R}_j)}\, y_{jp} \quad \forall\, p \in \mathcal{R}\}$, which is to say that the set of possible states of the system at the beginning of time period $t+1$ is obtained by using some feasible route assignment to ensure that all product purchases without a route assignment are assigned to a route. In this case, we can find the optimal policy by computing the value functions $(J_t : t \in \mathcal{T})$ through the dynamic program

$$J_t(\boldsymbol{x},\boldsymbol{z}) = \max_{S\subseteq\mathcal{J}} \left\{ \sum_{j\in\mathcal{J}} \phi_{jt}(S)\, \mathbf{1}_{(\mathcal{F}(\boldsymbol{x}+\boldsymbol{e}_j,\boldsymbol{z})\neq\varnothing)} \left\{ f_j + \max_{(\overline{\boldsymbol{x}},\overline{\boldsymbol{z}})\in\mathcal{G}_t(\boldsymbol{x}+\boldsymbol{e}_j,\boldsymbol{z})} J_{t+1}(\overline{\boldsymbol{x}},\overline{\boldsymbol{z}}) \;-\; \max_{(\overline{\boldsymbol{x}},\overline{\boldsymbol{z}})\in\mathcal{G}_t(\boldsymbol{x},\boldsymbol{z})} J_{t+1}(\overline{\boldsymbol{x}},\overline{\boldsymbol{z}}) \right\} \right\} + \max_{(\overline{\boldsymbol{x}},\overline{\boldsymbol{z}})\in\mathcal{G}_t(\boldsymbol{x},\boldsymbol{z})} J_{t+1}(\overline{\boldsymbol{x}},\overline{\boldsymbol{z}}), \tag{19}$$

with the boundary condition $J_{T+1}(\boldsymbol{x},\boldsymbol{z}) = 0$ for all $(\boldsymbol{x},\boldsymbol{z}) \in \mathbb{Z}_+^{|\mathcal{J}|+|\mathcal{R}|}$. The dynamic program is similar to (2), but the route assignments can change the state of the system from time period t to $t+1$.

Value Function Approximation and Approximate Policy:

Using $\psi_p(\boldsymbol{w}) = \min_{i\in A_p}\{w_i/c_i\}$ and letting the adjustable parameters $(\hat{\gamma}_{jpt} : j \in \mathcal{J},\ p \in \mathcal{R}_j,\ t \in \mathcal{T})$ be computed exactly as in (4)-(6), we consider value function approximations of the form

$$\hat{J}_t(\boldsymbol{x},\boldsymbol{z}) = \max_{(\boldsymbol{y},\boldsymbol{w})\in\mathbb{Z}_+^{\sum_{j\in\mathcal{J}}|\mathcal{R}_j|+|\mathcal{L}|}} \left\{ \sum_{j\in\mathcal{J}} \sum_{p\in\mathcal{R}_j} \hat{\gamma}_{jpt}\, \psi_p(\boldsymbol{w}) \;:\; \sum_{p\in\mathcal{R}_j} y_{jp} = x_j \quad \forall\, j \in \mathcal{J}, \quad \sum_{j\in\mathcal{J}} \sum_{p\in\mathcal{R}_j} a_{ip}\, y_{jp} + \sum_{p\in\mathcal{R}} a_{ip}\, z_p + w_i = c_i \quad \forall\, i \in \mathcal{L} \right\}. \tag{20}$$

The way we compute the value function approximation above is similar to the way we compute the value function approximation in (3), but the problem above takes into account the fact that we cannot change the route assignments for the product purchases that have already been irrevocably assigned to a route. In the value function approximation above, we still use the functional form $\psi_p(\boldsymbol{w}) = \min_{i\in A_p}\{\frac{w_i}{c_i}\}$ as before. Furthermore, we continue calibrating the adjustable parameters $(\hat{\gamma}_{jpt} : j \in \mathcal{J},\ p \in \mathcal{R}_j,\ t \in \mathcal{T})$ in the value function approximation above exactly as in (4)-(6). Although we calibrate the adjustable parameters $(\hat{\gamma}_{jpt} : j \in \mathcal{J},\ p \in \mathcal{R}_j,\ t \in \mathcal{T})$ as in (4)-(6), the way we compute the value function approximations in (20) is different from (3). We will be able to show that we can use the value function approximations in (20) to come up with a policy with a performance guarantee under periodic route assignments. Lastly, we give our approximate policy that is driven by our value function approximations. If the state of the system at the beginning of time period t is $(\boldsymbol{x},\boldsymbol{z})$, then our approximate policy offers the assortment of products

$$S_t^{\mathsf{App}}(\boldsymbol{x},\boldsymbol{z}) = \max_{S\subseteq\mathcal{J}} \left\{ \sum_{j\in\mathcal{J}} \phi_{jt}(S)\, \mathbf{1}_{(\mathcal{F}(\boldsymbol{x}+\boldsymbol{e}_j)\neq\varnothing)} \left\{ f_j + \max_{(\overline{\boldsymbol{x}},\overline{\boldsymbol{z}})\in\mathcal{G}_t(\boldsymbol{x}+\boldsymbol{e}_j,\boldsymbol{z})} \hat{J}_{t+1}(\overline{\boldsymbol{x}},\overline{\boldsymbol{z}}) \; - \max_{(\overline{\boldsymbol{x}},\overline{\boldsymbol{z}})\in\mathcal{G}_t(\boldsymbol{x},\boldsymbol{z})} \hat{J}_{t+1}(\overline{\boldsymbol{x}},\overline{\boldsymbol{z}}) \right\} \right\}.$$

Furthermore, after observing the product purchase at time period t, if the state of the system is $(\boldsymbol{x},\boldsymbol{z})$, then our approximate policy makes the route assignments in such a way that the state of the system at the beginning of time period $t+1$ is $Z_t^{\mathsf{App}}(\boldsymbol{x},\boldsymbol{z}) = \arg\max_{(\overline{\boldsymbol{x}},\overline{\boldsymbol{z}})\in\mathcal{G}_t(\boldsymbol{x},\boldsymbol{z})} \hat{J}_{t+1}(\overline{\boldsymbol{x}},\overline{\boldsymbol{z}})$. If time period t is not a route assignment period, then $\mathcal{G}_t(\boldsymbol{x},\boldsymbol{z}) = \{(\boldsymbol{x},\boldsymbol{z})\}$, which implies that $Z_t^{\mathsf{App}}(\boldsymbol{x},\boldsymbol{z}) = (\boldsymbol{x},\boldsymbol{z})$. Thus, our approximate policy follows the greedy action with respect to the value function approximations to decide which assortment to offer. After observing the product purchase, if we are at a route assignment period, then our approximate policy maximizes the value function approximation to make the route assignment decisions. In the next theorem, we give a performance guarantee for our approximate policy under periodic route assignments. We defer the proof to Appendix C. The proof digs into the properties of the problem $\max_{(\overline{\boldsymbol{x}},\overline{\boldsymbol{z}})\in\mathcal{G}_t(\boldsymbol{x},\boldsymbol{z})} \hat{J}_{t+1}(\overline{\boldsymbol{x}},\overline{\boldsymbol{z}})$, using which we lower bound the total expected revenue of our approximate policy.

Theorem 6.1 (Performance Guarantee under Periodic Route Assignments) *The total expected revenue obtained by the approximate policy under periodic route assignments is at least* $1/(1+\theta L)$ *fraction of the optimal total expected revenue.*

In our computational experiments, we will use our approximate policy to explore the revenue implications of making irrevocable route assignments during the course of the selling horizon.

7. Approximate Policies with Different Route Assignment Frequencies

We argue that an approximate policy constructed under the assumption that we make the route assignments at every time period can still provide the theoretical performance guarantee of $1/(1+L)$ when we make the route assignments at the end of the selling horizon. Notwithstanding the theoretical performance guarantee, however, we also demonstrate that switching the timing of the route assignment can result in substantial revenue loss in practice. In particular, if we make the route assignments less frequently, then we can obtain a larger total expected revenue. Thus, letting $\mathsf{OPT}^{\mathcal{A}}$ be the optimal total expected revenue when we make the route assignments at the set of time periods $\mathcal{A}$, if $\mathcal{A}\subseteq\mathcal{B}$, then we have $\mathsf{OPT}^{\mathcal{A}} \geq \mathsf{OPT}^{\mathcal{B}}$. Also, the linear program in (8) is formulated under the assumption that the arrivals and choices of the customers take on their expected values without specifying when we make the route assignments, so the optimal objective value of problem (8) provides an upper bound on the optimal total expected revenue irrespective of the set of time periods at which we make the route assignments. Therefore, noting that Z^*_{LP} is the optimal objective value of problem (8), we have $Z^*_{\mathsf{LP}} \geq \mathsf{OPT}^{\mathcal{A}}$ for any $\mathcal{A} \subseteq \mathcal{T}$. Lastly, the proof of Theorem 6.1 is based on showing that the total expected revenue of our approximate policy is at least $1/(1+\theta L)$ fraction of the optimal objective value of the linear program in (8). Thus, using $\mathsf{App}^{\mathcal{A}}$ to denote the total expected revenue of the approximate policy when we make the route assignments at the set of time periods $\mathcal{A}$, we have $\mathsf{App}^{\mathcal{A}} \geq \frac{1}{1+\theta L} Z^*_{\mathsf{LP}}$ for any $\mathcal{A} \subseteq \mathcal{T}$. We consider the two cases, the first one making the route assignments at every time period and the second one making the route assignments only at the end of the selling horizon. The optimal total expected revenues in the two cases are, respectively, $\mathsf{OPT}^{\mathcal{T}}$ and $\mathsf{OPT}^{\{T\}}$. Since $\{T\} \subseteq \mathcal{T}$, we have $\mathsf{OPT}^{\{T\}} \geq \mathsf{OPT}^{\mathcal{T}}$. Also, since $Z^*_{\mathsf{LP}} \geq \mathsf{OPT}^{\mathcal{A}}$ and $\mathsf{App}^{\mathcal{A}} \geq \frac{1}{1+\theta L} Z^*_{\mathsf{LP}}$ for any $\mathcal{A} \subseteq \mathcal{T}$, we get $\mathsf{App}^{\mathcal{T}} \geq \frac{1}{1+\theta L} Z^*_{\mathsf{LP}} \geq \frac{1}{1+\theta L} \mathsf{OPT}^{\{T\}}$.

Having $\mathsf{App}^{\mathcal{T}} \geq \frac{1}{1+\theta L} Z^*_{\mathsf{LP}} \geq \frac{1}{1+\theta L} \mathsf{OPT}^{\{T\}}$ implies that the approximate policy computed under the assumption that we make the route assignments at every time period still provides a performance guarantee of $1/(1+\theta L)$ for a problem with flexible resources that allows delaying the route assignments to the end of the selling horizon. If we make the route assignments at every time period, then it is enough to keep the remaining capacity of each resource in the state variable, in which case, we can use existing approximate policies for network revenue management problems without flexible resources. This observation points out that if we have a problem instance that allows delaying the route assignments to the end of the selling horizon, then we can still obtain a performance guarantee of $1/(1+\theta L)$ by using an approximate policy that makes the route assignments at every time period. Nevertheless, we make three observations.

First, for a problem that allows delaying the route assignments to the end of the selling horizon, the *practical* performance of a policy that makes the route assignments at every time period can be quite poor. We shortly give one example. Second, given that it is practically useful to delay the route assignments to the end of the selling horizon, it was not clear whether we can construct a policy that delays the route assignments while maintaining the performance guarantee of $1/(1+\theta L)$. Our approximate policies demonstrate that we can. Third, the observation in the previous paragraph occurs because we establish our results by comparing the performance of our policies with the deterministic benchmark Z^*_{LP}, which is agnostic to the timing of the route assignments. Similar situation occurs when we have random resource usage durations, but a deterministic benchmark is agnostic to when the usage durations are revealed to the decision maker.

We give a problem instance that allows delaying the route assignments to the end of the selling horizon, but making the route assignments at every time period loses at least half of the optimal total expected revenue. The set of resources is $\mathcal{L}$. We use $K = |\mathcal{L}|$ to denote the number of resources. Each resource has one unit of capacity. The set of products is $\mathcal{J} = \mathcal{L} \cup \{\phi\}$. For each $j \in \mathcal{L}$, the revenue of product j is $f_j = K$. The revenue of product ϕ is $f_\phi = K-1$. For each $j \in \mathcal{L}$, we can use only the route $\{j\}$ to serve product j, so $\mathcal{R}_j = \{\{j\}\}$. Thus, these products are inflexible. We can use any route with exactly $K-1$ resources to serve product ϕ, so $\mathcal{R}_\phi = \{Q \subseteq \mathcal{L} : |Q| = K-1\}$. There are two time periods. The customer arriving at the first time period requests product ϕ with probability one. The customer arriving at the second time period requests product $j \in \mathcal{L}$ with probability $1/K$. We can delay the route assignments to the end of the selling horizon.

It is optimal to accept the product requests at both time periods, yielding a total expected revenue of $K-1+K = 2K-1$. At the end of the selling horizon, if the product request from the second time period was for product j, then we assign the product request from the first time period to route $\mathcal{L} \setminus \{j\}$ and assign the product request from the second time period to route $\{j\}$. Consider the case where we make the route assignments at every time period. If we accept the product request at the first time period and assign it to route $\mathcal{L} \setminus \{j\}$ for some $j \in \mathcal{L}$, then we can only accept a request for product j at the second time period, yielding a total expected revenue of $f_\phi + \frac{1}{K} f_j = K - 1 + \frac{1}{K} K = K$. If we reject the product request at the first time period, then we can accept any request at the second time period, yielding a total expected revenue of K. In either case, the total expected revenue is K. We have $\lim_{K\to\infty} \frac{2K-1}{K} = 2$, so making route assignments at every time period loses at least half of the total expected revenue as K gets large.

Thus, a policy that makes the route assignments at every time period may have a theoretical performance guarantee even when we can delay the route assignments to the end of the selling horizon, but the practical performance of this policy can be poor. In Appendix D, we show that the linear program in (8) can be transformed into an equivalent linear program without flexible products, providing an alternative argument that a policy computed under limited flexibility may give a performance guarantee for network revenue management with flexible products.

8. Computational Experiments

We give two sets of computational experiments. The first one is on providing at-home services, whereas the second one is on bipartite matching.

8.1 At-Home Service Provider

We describe our experimental setup, followed by our benchmark policies and computational results. We also investigate the benefit from making the route assignments with different frequencies.

Experimental Setup: We consider a company providing at-home services, such as cleaning, pet walking or plant care, in hourly blocks. Some customers would like to receive service at a fixed time, whereas others are flexible, deferring the choice to the company in return for a discount. We focus on a particular day of service. The resources correspond to one-hour blocks. There are eight hours in the day and services start and end at the beginning of an hour. Thus, the set of resources is $\mathcal{L}=\{1,\ldots,8\}$, where resource ℓ is the service capacity during hour ℓ. Services purchased by the customers have two dimensions. First, customers can purchase service for one or two hours. Second, customers can purchase service starting at a fixed time or at a flexible time in the morning, in the afternoon or throughout the whole day. In the last three cases, the company chooses the time of service. We use the pair $(d,[\ell,k])$ to denote a product, where d is the duration of service and $[\ell,k]$ is set of possible starting times for service. Thus, the set of products is $\mathcal{J}=\{(d,[\ell,k]):d=1,2,\;[\ell,k]=[1,1],[2,2],\ldots,[8,8],[1,4],[5,8],[1,8]\}$, where, for example, the product $(d,[\ell,\ell])$ corresponds d hours of service starting at fixed hour ℓ and the product $(d,[1,4])$ corresponds d hours of service starting at a flexible time in the morning. If a customer purchases the product $(d,[\ell,\ell])$, then the only route to serve the customer includes the resources $\{\ell,\ell+d-1\}$. If a customer purchases the product $(d,[1,4])$, then there are four routes to serve the customer, each route including the set of resources $\{\ell,\ell+d-1\}$ for $\ell=1,\ldots,4$. The revenues of the products of the form $(d,[\ell,\ell])$, $(d,[1,4])$, $(d,[5,8])$ and $(d,[1,8])$ are, respectively, $d\times 80$, $\beta\, d\times 80$, $\beta\, d\times 80$ and $\beta^2 d\times 80$, where β is the discount factor for being flexible. We vary β.

In our model and technical results, the customers arriving into the system at a particular time period choose among the offered products according to the same choice model, but it is simple to extend our work to the case where there are multiple customer types and customers of different types choose according to different choice models. In our computational experiments, we have a total of 18 customer types, 16 of them are inflexible and two are flexible. We index the inflexible customer types by $\mathcal{C}_{\mathsf{Fixed}}=\{(d,\ell):d=1,2,\;\ell=1,2,\ldots,8\}$, where an inflexible customer of type (d,ℓ) is interested in receiving service for d hours starting at the fixed hour ℓ. If product $(d,[\ell,\ell])$ is made available to this customer, then she purchases. Otherwise, she leaves without a purchase. We index the flexible customer types by $\mathcal{C}_{\mathsf{Flex}}=\{(d,\varnothing):d=1,2\}$, where a flexible customer of type $(d,\varnothing)$ is interested in receiving service for d hours, but she is not keen on the time of service. She makes a choice within the set of products $(d,[1,4])$, $(d,[5,8])$ and $(d,[1,8])$. We use the multinomial logit model to capture the choice process of the flexible customers. Using v_j^d to denote the preference weight that a flexible customer of type $(d,\varnothing)$ attaches to product $j\in\{(d,[1,4]),(d,[5,8]),(d,[1,8])\}$ and v_0^d to denote the preference weight of the no-purchase option, if such a customer is offered the assortment of products $S\subseteq\{(d,[1,4]),(d,[5,8]),(d,[1,8])\}$, then she purchases product j in the assortment with probability $v_j^d/(v_0^d+\sum_{k\in S}v_k^d)$. We generate the preference weights for the products by sampling them from the uniform distribution over $[0,5]$. We calibrate the preference weight of the no-purchase option so that if we offer all products $\{(d,[1,4]),(d,[5,8]),(d,[1,8])\}$ to a flexible customer, then she leaves without a purchase with probability P_0. We vary P_0.

There are $T=100$ time periods in the selling horizon. Using $\{(d,\ell): d=1,2, \ell=\varnothing,1,2,\ldots,8\}$ to index all customer types, at time period t, a customer of type (d,ℓ) arrives into the system with probability $\lambda_{(d,\ell),t}$. We calibrate the arrival probabilities in such a way that the arrival probabilities of inflexible customers increases over time, whereas the arrival probabilities of flexible customers decrease. In this way, it becomes important to carefully reserve the capacity for the inflexible customers that tend to arrive later in the selling horizon. Lastly, we proceed as follows to generate the available capacity for each resource. If all inflexible customers make a purchase for the product they are interested in, then the total expected demand for resource ℓ from the inflexible customers is $\sum_{t\in\mathcal{T}}\sum_{(d,k)\in\mathcal{C}_{\mathsf{Fixed}}} \mathbf{1}_{(k\le \ell\le k+d-1)}\,\lambda_{(d,\ell),t}$. For each of the flexible customer types, we consider offering the full assortment of products $\{(d,[1,4]),(d,[5,8]),(d,[1,8])\}$ that such customers are interested in. Assuming that if a flexible customer makes a purchase, then we assign the customer to one of the routes for the purchased product with equal probability, we compute the total expected demand for resource ℓ from the flexible customers. Letting Demand_ℓ be the total expected demand for resource ℓ from the inflexible and flexible customers, we set the capacity of resource ℓ as $\lfloor \mathsf{Demand}_\ell/\alpha\rfloor$, where α controls the tightness of the capacities. We also vary α.

Varying $\beta\in\{0.8,0.9\}$, $P_0\in\{0.1,0.4,0.6\}$ and $\alpha\in\{1.2,1.4,1.6\}$, we obtain 18 parameter configurations. For each one, we generate a test problem as in the previous three paragraphs.

Benchmark Policies: We use three benchmarks. The first benchmark is the approximate policy in Section 4. We refer to this benchmark as AP, standing for approximate policy. The second benchmark uses the linear program in (8) to estimate the value of a unit of resource, which is called the bid price. We refer to this benchmark as BP, standing for bid price policy. The third benchmark is the approximate policy in Section 6, where we make the route assignments at each time period. We refer to this benchmark as NF, standing for no flexibility. We detail each benchmark. In our construction of the value function approximations, we use a tuning parameter θ. We get the strongest performance guarantee with $\theta=1$, but setting $\theta=1$ may not lead to the best numerical performance. To choose the value of θ, we consider the values in the interval $[1,8)$ in increments of 0.1, yielding 70 possible values. Using $\{\theta^k: k=1,\ldots,70\}$ to capture these values, we compute the value function approximations used by AP under each value of θ^k for $k=1,\ldots,70$ and simulate the performance of the corresponding policy. Given that we set the tuning parameter as θ^k, we use Rev_t^k to denote the total expected revenue obtained by AP over time periods $t,\ldots,T$. In AP, we split the selling horizon to five equal segments. At the beginning of segment q, which is time period $(q-1)\frac{T}{5}+1$, we switch to using the value function approximation computed with the value of the tuning parameter $\theta = \arg\max_{k=1,\ldots,70} \mathsf{Rev}^k_{(q-1)\frac{T}{5}+1}$. Adjusting the value of the tuning parameter in this fashion improves the performance of AP by within a percentage point.

Considering BP, the second constraint in (8) ensures that we do not violate the resource capacities. Letting $(\mu_i^* : i \in \mathcal{L})$ be the optimal values of the dual variables for these constraints, we use μ_i^* to capture the value of a unit of resource i. For each product j, we choose an ideal route given by $\arg\max_{p\in\mathcal{R}_j} f_j - \sum_{i\in\mathcal{L}} a_{ip}\, \mu_i^*$, where we maximize the revenue from the product net of the values of the resources used by the route. Letting $\overline{f}_j$ be the optimal objective value of the last problem, $\overline{f}_j$ is the net revenue from product j after adjusting for the values of the resources in the ideal route. If the state of the system at time period t is $\boldsymbol{x}$, then BP offers the assortment of products $\arg\max_{S\subseteq\mathcal{J}} \sum_{j\in\mathcal{J}} \phi_{jt}(S)\, \mathbf{1}_{(\mathcal{F}(\boldsymbol{x}+\boldsymbol{e}_j)\neq\varnothing)}\, \overline{f}_j$, which is the assortment that maximizes the expected net revenue. Our discussion of BP so far has been for the case with a single customer type, but extension to multiple customer types requires minor adjustments. In BP, we split the selling horizon into five equal segments and re-compute the bid prices at the beginning of each segment. In particular, if the state of the system at the beginning of segment q is $\boldsymbol{x}$, then we replace the set of time periods with $\{(q-1)\frac{T}{5}+1,\ldots,T\}$ and add x_j to the left side of the first constraint in (8). We use the optimal values of the dual variables for the second constraint as bid prices.

In NF, we make the route assignments at each time period, so this benchmark does not take advantage of the flexibility provided by delaying the route assignments.

Computational Results: We give our results in Table 1. The first column shows the parameters (β, P_0, α) for each test problem. To estimate the total expected revenues obtained by AP, BP and NF, we simulate their performance for 100 sample paths. Recalling that the optimal objective value of the linear program in (8) is an upper bound on the optimal total expected revenue, the second to fourth columns give the total expected revenues obtained by AP, BP and NF expressed as a percentage of the upper bound. Our results indicate that AP performs significantly better than BP for our test problems. Over all test problems, the average percent gap between the total expected revenues obtained by AP and BP is 5.46%. There are test problems where the performance gap between the two benchmarks reaches 9.98%. The performance gap between AP and BP tends to go up as α increases so that the resource capacities get tighter. For the test problems with $\alpha = 1.2, 1.4$ and 1.6, the average percent gaps between the total expected revenues of AP and BP are, respectively, 4.19%, 5.72% and 6.48%. When the resource capacities are tight, it is especially important to reserve the capacity for the inflexible customers that tend to arrive later. It appears that AP is able to do a better job of reserving the capacity for the inflexible customers.

Comparing AP with NF, AP consistently improves the performance of NF. Over all test problems, the average percent gap between the total expected revenues obtained by AP and NF

Params. (β, P_0, α)	AP	BP	NF	Params. (β, P_0, α)	AP	BP	NF
$(0.9, 0.1, 1.2)$	94.11	90.94	91.56	$(0.8, 0.1, 1.2)$	92.93	88.16	90.94
$(0.9, 0.1, 1.4)$	94.18	88.05	92.55	$(0.8, 0.1, 1.4)$	93.37	88.38	91.73
$(0.9, 0.1, 1.6)$	94.07	90.36	92.81	$(0.8, 0.1, 1.6)$	92.76	89.61	91.69
$(0.9, 0.4, 1.2)$	91.58	86.02	88.61	$(0.8, 0.4, 1.2)$	90.03	87.19	87.34
$(0.9, 0.4, 1.4)$	92.03	87.3	89.52	$(0.8, 0.4, 1.4)$	90.37	86.05	88.07
$(0.9, 0.4, 1.6)$	92.38	85.26	90.44	$(0.8, 0.4, 1.6)$	90.62	84.97	88.83
$(0.9, 0.6, 1.2)$	88.76	84.94	85.17	$(0.8, 0.6, 1.2)$	87.37	84.67	84.02
$(0.9, 0.6, 1.4)$	89.44	83.05	86.66	$(0.8, 0.6, 1.4)$	87.28	82.54	84.50
$(0.9, 0.6, 1.6)$	90.18	81.17	86.99	$(0.8, 0.6, 1.6)$	87.78	81.11	85.50
Avg.	91.86	86.34	89.37	Avg.	90.28	85.85	88.07

Table 1 Computational results for the at-home service provider setting.

is 2.65%. There are test problems where the performance gap between AP and NF exceeds 4%. Such revenue improvements is considered quite significant in the revenue management setting as a few percentage point increase in the total expected revenue translates into much larger increase in the total expected profit. Considering that NF is a version of AP without flexibility, the flexibility offered by AP can yield noticeable improvements in the total expected revenues.

In Section 6, we show that we can use our model to make the route assignments periodically. Delaying the route assignments to the end of the selling horizon provides more flexibility, resulting in higher total expected revenues, but making the route assignments periodically lets the customers know about their hour of service earlier, resulting in better service. In Table 2, we focus on one test problem with $(\beta, P_0, \alpha) = (0.9, 0.6, 1.2)$ and give the total expected revenue obtained by our approximate policy when we make the route assignments every κ time periods. Setting $\kappa = 100$ delays the route assignments to the end of the selling horizon, corresponding to AP, whereas setting $\kappa = 1$ lets each customer know about her route assignment just after her purchase, corresponding to NF. We vary $\kappa \in \{1, 10, 20, 30, 40, 50, 100\}$. The first row shows the value of κ. Letting Base be the total expected revenue obtained with $\kappa = 1$, the second row shows the percent gap between the total expected revenues obtained with a particular value of κ and Base. Compared with making the route assignments immediately for each customer, maintaining maximum amount of flexibility and delaying the route assignments to the end of the selling horizon provides an improvement of 4.22% in the total expected revenue. Thus, there is significant value in delaying the route assignments as much as possible. Noting the data point with $\kappa = 50$, introducing just one more time point at which we make routing assignments in the middle of the selling horizon reduces the percent gap with Base to 1.40%. For this test problem, maintaining full flexibility and delaying route assignments as much as possible has significant benefits. We report results for one test problem, but we observed similar behavior in others. Thus, for our test problems, maintaining full flexibility appears to be important. Using our model, one can quantify similar tradeoffs in other settings.

κ	1	10	20	30	40	50	100
% Gap	0.00	0.25	0.62	1.17	1.35	1.40	4.22

Table 2 Changes in the total expected revenue with periodic route assignments when compared with $\kappa = 1$.

8.2 Bipartite Matching

We consider a bipartite matching problem with flexible jobs, where we can delay the decision of exactly which resource to use to serve an accepted job.

Experimental Setup: We give an overview of the bipartite matching problem. The set of resources is $\mathcal{L}$. Resource i has a capacity of c_i. The set of job types is $\mathcal{J}$. The revenue associated with a job of type j is f_j. A job of type j is compatible with the resources in the set $\mathcal{L}_j \subseteq \mathcal{L}$. In other words, if we accept a job of type j, then we have to use one of the resources in $\mathcal{L}_j$ to serve the job. There are T time periods in the planning horizon indexed by $\mathcal{T} = \{1, \ldots, T\}$. At time period t, a job of type j arrives with probability λ_{jt}. We can accept this job only if there is a feasible way to assign all accepted jobs so far to compatible resources. We make the resource assignment decisions at the end of the planning horizon. The goal is to find a policy to decide whether to accept each job so that we maximize the total expected revenue, while ensuring that we always have a feasible way to assign all accepted jobs to compatible resources. In Appendix E, we give a formulation for the bipartite matching problem with flexible jobs and explain that we can use our approach to construct an approximate policy with a performance guarantee. We work with the bipartite matching problem because this problem allows us to easily adjust the level of inherent flexibility by varying the numbers of resources compatible with jobs of different types.

We generate instances of the bipartite matching problem by using the following approach. We have 100 resources and 100 job types. The number of resources that jobs of each type are compatible with varies between 3 and 12. In other words, we have $3 \leq |\mathcal{L}_j| \leq 12$ for all $j \in \mathcal{J}$. For each job type j, we sample the elements of $\mathcal{L}_j$ uniformly within the set of resources $\mathcal{L}$. The revenue associated with a job of type j is given by $f_j = 1/\log(2 + |\mathcal{L}_j|)$. Thus, if a job type is compatible with a larger number of resources, then its revenue is smaller. For a job type j, if $|\mathcal{L}_j| > 3$, then we refer to this job type as a flexible job type, whereas if $|\mathcal{L}_j| = 3$, then we refer to this job type as an inflexible job type. There are $T = 1000$ time periods. We generate the the arrival probabilities for different job types in such a way that flexible jobs tend to arrive earlier in the planning horizon, which necessitates protecting the resource capacities carefully for the inflexible jobs that tend to arrive later. In particular, we let $g(t) = \frac{T-t+1}{T}$ and $h(t) = \frac{t-1}{4T}$. Using n and m to, respectively, denote the numbers of flexible and inflexible jobs, if job type j is flexible, then we set $\lambda_{jt} = g(t)/(n\, g(t) + m\, h(t))$, whereas if job j is inflexible, then we set $\lambda_{jt} = h(t)/(n\, g(t) + m\, h(t))$. Thus, $\sum_{j \in \mathcal{J}} \lambda_{jt} = 1$, so the total expected number of job arrivals is $T = 1000$. We set the capacity of each resource i as $c_i = 10$, so the total resource capacity matches the total expected number of job arrivals. Using this approach, we generate one problem instance, referred to as the base instance.

We gradually increase the numbers of compatible resources for the different job types in the base problem to understand the benefit from having more flexible job types.

Computational Results: Building on the base instance, we generate six more problem instances indexed by $\eta \in \{1, \ldots, 6\}$. In instance η, each job type has η more compatible resources when compared with the base instance. In particular, using $\mathcal{L}_j^0$ and $\mathcal{L}_j^\eta$ to, respectively, denote the sets of compatible resources for job type j in the base instance and instance η, we have $\mathcal{L}_j^\eta \supseteq \mathcal{L}_j^0$ and $|\mathcal{L}_j^\eta \setminus \mathcal{L}_j^0| = \eta$. We vary the degree of flexibility in a problem instance along two dimensions. First, we vary $\eta \in \{1, \ldots, 6\}$. Thus, if η is larger, then we have more compatible resources for each job type. Second, we use our approximate policy by making the resource assignment decisions every κ time periods. We vary $\kappa \in \{1, 10, 50, 100, 500, 1000\}$. Therefore, if κ is larger, then we delay the resource assignment decisions further. We give our results in Table 3. The first column shows the value of κ, controlling the frequency of the resource assignment decisions. The next seven columns show the total expected revenues obtained by our approximate policy for problem instance $\eta \in \{0, 1, \ldots, 6\}$, where we use $\eta = 0$ to denote the base instance. The last six columns show the percent gaps between the total expected revenues obtained by our approximate policy for instance η and the base instance. In other words, letting Rev^η be the total expected revenue for instance η, the last six columns show $100 \times \frac{\mathsf{Rev}^\eta - \mathsf{Rev}^0}{\mathsf{Rev}^0}$ for $\eta \in \{1, \ldots, 6\}$.

Our approximate policy, as expected, obtains larger total expected revenues as the numbers of compatible resources for the different job types increase. Compared to the base instance, on average, adding one more compatible resource for each job type increases the total expected revenue by 2.71%, whereas adding six more compatible resources for each job type increases the total expected revenue by 9.53%. Increasing the number of resources compatible with each job type is one way of increasing the flexibility and delaying the resource assignment decisions is another way. Our results allow us to assess the effectiveness of the two approaches. For example, considering the base instance, if we make the resource assignments at the end of the planning horizon instead of at each time period, then the total expected revenue increases from 551.71 to total expected revenue of 570.37, but adding two more compatible resources for each job type provides a similar increase in the total expected revenue, increasing the total expected revenue from 551.71 to 574.10. Naturally, while one of these sources of flexibility may be available in some operational settings, the other may not be. Lastly, over all of our instances, making the resource assignment decisions at the end of the planning horizon instead of at the end of each time period, on average, increases the total expected revenue by 5.21%.

	Total Expected Revenue							% Gap with Base Instance					
κ	0	1	2	3	4	5	6	1	2	3	4	5	6
1	551.71	563.95	574.10	573.97	578.06	590.07	603.21	2.22	4.06	4.04	4.78	6.95	9.34
10	552.25	563.14	573.27	573.76	579.14	590.68	603.32	1.97	3.81	3.89	4.87	6.96	9.25
50	551.62	563.15	573.56	574.32	579.24	591.28	604.84	2.09	3.98	4.12	5.01	7.19	9.65
100	552.48	564.54	574.09	574.50	580.20	592.55	605.72	2.18	3.91	3.99	5.02	7.25	9.64
500	562.15	581.84	591.20	596.57	607.45	618.53	620.02	3.50	5.17	6.12	8.06	10.03	10.29
1000	570.37	594.90	607.86	610.59	618.26	621.64	621.64	4.30	6.57	7.05	8.40	8.99	8.99
Avg.								2.71	4.58	4.87	6.02	7.90	9.53

Table 3 Computational results for the bipartite matching setting.

9. Conclusions

We gave a policy with a performance guarantee for network revenue management problems with flexible products. They key ingredient in our approach is to solve an optimization problem to convert a value function approximation that is defined as a function of the remaining resource capacities into a value function that is defined as a function of the numbers of product purchases. Such a conversion idea has not been used in the literature to obtain performance guarantees and it allows us to give the first policy with a performance guarantee that can delay the route assignments. The performance guarantee holds irrespective of whether we make the route assignments at the end of the selling horizon, right after each product purchase or periodically over the selling horizon. It turns out that our conversion idea has implications for other problems. In Appendix E, we give problems in bipartite matching, joint assortment and price optimization and managing reusable resources, where we can use our conversion idea to incorporate flexible products. We propose three avenues for future research. First, we used our model to numerically check the benefit from making the route assignments with different frequencies, but it would be useful to give upper and lower bounds on the benefit from the delaying route assignments, quantifying the tradeoff between being customer-centric through frequent route assignments and firm-centric through delayed route assignments. Even a stylized model to investigate such tradeoffs would be interesting. Second, we solve the integer program in (3) to compute our value function approximations. In Appendix F, we give one application setting where we can solve problem (3) through a dynamic program. It is interesting to explore other approaches to compute our value function approximations, even if these approaches are restricted to special application settings. Third, the necessity to check whether $\mathcal{F}(\boldsymbol{x}) \neq \varnothing$ is an inherent part of our dynamic programming formulation of the problem, rather than our specific approximate policy. It is NP-complete to carry out this check. Another interesting research avenue is to study alternative formulations and application settings where this check is not necessary or can be carried out in polynomial time.

Acknowledgements: We thank the department editor, associate editor and two referees for their constructive comments. The material in Appendix D is provided by one of the referees.

References

Alaei, S., M. Hajiaghayi, V. Liaghat. 2012. Online prophet-inequality matching with applications to ad allocation. *Proceedings of the 13th ACM Conference on Electronic Commerce*. EC 2012, Association for Computing Machinery, New York, NY, USA, 18–35.

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

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

Cheung, W. C., D. Simchi-Levi. 2016. Efficiency and performance guarantees for choice-based network revenue management problems with flexible products. Tech. rep., MIT, Cambridge, MA.

Elmachtoub, A. N., M. L. Hamilton. 2021. The power of opaque products in pricing. *Management Science* **67**(8) 4686–4702.

Elmachtoub, A. N., Y. Wei, Y. Zhou. 2015. Retailing with opaque products. Tech. rep., Department of Industrial Engineering and Operations Research, Columbia University, New York, NY.

Elmachtoub, A. N., D. Yao, Y. Zhou. 2019. The value of flexibility from opaque selling. Tech. rep., Department of Industrial Engineering and Operations Research, Columbia University, New York, NY.

Erdelyi, A., H. Topaloglu. 2010. A dynamic programming decomposition method for making overbooking decisions over an airline network. *INFORMS Journal on Computing* **22**(3) 443–456.

Fay, S., J. Xie. 2008. Probabilistic goods: A creative way of selling products and services. *Marketing Science* **27**(4) 674–690.

Fay, S., J. Xie. 2015. Timing of product allocation: Using probabilistic selling to enhance inventory management. *Management Science* **61**(2) 474–484.

Feng, Y., R. Niazadeh, A. Saberi. 2022. Near-optimal Bayesian online assortment of reusable resources. Tech. rep., The University of Chicago, Booth School of Business, Chicago, IL.

Gallego, G., G. Iyengar, R. Phillips, A. Dubey. 2004. Managing flexible products on a network. Computational Optimization Research Center Technical Report TR-2004-01, Columbia University.

Gallego, G., R. Phillips. 2004. Revenue management with flexible products. *Manufacturing & Service Operations Management* **6**(4) 321–337.

Gallego, G., C. Stefanescu. 2009. Upgrades, upsells and pricing in revenue management. Tech. rep., Columbia University.

Gonsch, J., S. Koch, C. Steinhardt. 2014. Revenue management with flexible products: The value of flexibility and its incorporation into DLP-based approaches. *International Journal of Production Economics* **153** 280–294.

Hazan, E., S. Safra, O. Schwartz. 2006. On the complexity of approximating k-set packing. *Computational Complexity* **15**(1) 20–39.

Karaesmen, I., G. van Ryzin. 2004. Overbooking with substitutable inventory classes. *Operations Research* **52**(1) 83–104.

Koch, S., J. Gonsch, C. Steinhardt. 2017. Dynamic programming decomposition for choice-based revenue management with flexible products. *Transportation Science* **51**(4) 1046–1062.

Ma, W., D. Simchi-Levi, J. Zhao. 2021. Dynamic pricing (and assortment) under a static calendar. *Management Science* **67**(4) 2292–2313.

Ma, Y., P. Rusmevichientong, M. Sumida, H. Topaloglu. 2020. An approximation algorithm for network revenue management under nonstationary arrivals. *Operations Research* **68**(3) 834–855.

Manshadi, V., S. Rodilitz. 2022. Online policies for efficient volunteer crowdsourcing. *Management Science* **68**(9) 6572–6590.

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

Shumsky, R. A., F. Zhang. 2009. Dynamic capacity management with substitution. *Operations Research* **57**(3) 671–684.

Steinhardt, C., J. Gonsch. 2012. Integrated revenue management approaches for capacity control with planned upgrades. *European Journal of Operational Research* **223**(2) 380–391.

Xiao, Y., J. Chen. 2014. Evaluating the potential effects from probabilistic selling of similar products. *Naval Research Logistics* **61**(8) 604–620.

Xu, H., D. Yao, S. Zheng. 2011. Optimal control of replenishment and substitution in an inventory system with nonstationary batch demand. *Production and Operations Management* **20**(5) 727–736.

Yu, Y., X. Chen, F. Zhang. 2015. Dynamic capacity management with general upgrading. *Operations Research* **63**(6) 1372–1389.