

Revenue Management with Heterogeneous Resources: Unit Resource Capacities, Advance Bookings, and Itineraries over Time Intervals

Paat Rusmevichientong^{1*}, Mika Sumida¹, Huseyin Topaloglu², Yicheng Bai²

¹ Marshall School of Business, University of Southern California, Los Angeles, CA 90089

² School of Operations Research and Information Engineering, Cornell Tech, New York, NY 10044
rusmevic@marshall.usc.edu, mikasumi@marshall.usc.edu, ht88@cornell.edu, yb279@cornell.edu

August 5, 2021

We study revenue management problems with heterogeneous resources, each with unit capacity. An arriving customer makes a booking request for a particular interval of days in the future. We offer an assortment of resources in response to each booking request. The customer makes a choice within the assortment to use the chosen resource for her desired interval of days. The goal is to find a policy that determines an assortment of resources to offer to each customer to maximize the total expected revenue over a finite selling horizon. The problem has two useful features. First, each resource is unique with unit capacity. Second, each customer uses the chosen resource for a number of consecutive days. We consider static policies that offer each assortment of resources with a fixed probability. We show that we can efficiently perform rollout on any static policy, allowing us to build on any static policy and construct an even better policy. Next, we develop two static policies, each of which is derived from linear and polynomial approximations of the value functions. We give performance guarantees for both policies, so the rollout policies based on these static policies inherit the same guarantee. Lastly, we develop an approach for computing an upper bound on the optimal total expected revenue. Our results for efficient rollout, static policies, and upper bounds all exploit the aforementioned two useful features of our problem. We use our model to manage hotel bookings based on a dataset from a real-world boutique hotel, demonstrating that our rollout approach can provide remarkably good policies and our upper bounds can significantly improve those provided by existing techniques.

1. Introduction

Revenue management problems focus on managing limited service capacities to serve booking requests that arrive randomly over time. Serving a booking request generates a certain amount of revenue and consumes the availability of multiple types of service capacities. These problems appear in airlines, hospitality, retail, railways, and broadcasting, where the meanings of a service capacity and a booking request take different forms depending on the specific industry setting. The main tradeoff is between serving a current booking request to generate immediate revenue and reserving the service capacities for a more profitable booking request that may arrive in the future. Different booking requests consume different types of service capacities, so computing an optimal policy requires keeping track of all types of remaining service capacities simultaneously, resulting in the curse of dimensionality as the number of different types of service capacities increases.

We study revenue management problems with unique resources, each with unit capacity. An arriving customer makes a booking request for a particular interval of days in the future. We offer

*The first three authors are listed alphabetically. The last author mainly contributed to computational experiments.

an assortment of resources in response to each booking request. The customer makes a choice within the assortment to use the chosen resource for her desired interval of days and returns the resource after her use. The goal is to find a policy for determining an assortment of resources to offer to each customer to maximize the total expected revenue. Dynamic programming formulation of the problem has a high-dimensional state variable to keep track of the availability of resources on each day in the booking horizon, so it is computationally difficult to find the optimal policy. We give efficiently computable policies with performance guarantees and upper bounds on the optimal total expected revenue. As we discuss in our contributions below, all of our results exploit the fact that the resources have unit capacity and the customers use resources over consecutive days.

The class of revenue management problems that we consider in this paper appears in a number of applications. Market places for lodging, such as Airbnb and Vrbo, as well as boutique hotels and bed-and-breakfasts, offer unique rooms, apartments or houses. Customers make booking requests to use such lodging options for a number of consecutive days. Matching platforms for freelancers, such as Upwork and Fiverr, recommend differentiated workers with unique characteristics. Employers seek to make use of the skills of the workers for certain durations of time. When the projects are intensive enough that each freelancer can work on one project at a time, the problem of offering differentiated workers to employers also has the characteristics of our revenue management problem. With these applications in mind, our work is partly motivated by the boutique hotel Villa Mahal (2020) in Kalkan province of Turkey. The hotel offers six rooms: Moonlight Room, Moonlight Deluxe, Sunset Deluxe, Sunset Suite, Pool Room, and Cliff House. Each room is unique, as it is decorated differently, has different views, and offered at a different price. During the booking process, the customers are shown the available rooms for their desired days of stay, and they choose a specific room. Boutique Homes (2020) lists 150 boutique hotels with a similar setup.

Main Contributions: Our technical results focus on policies with performance guarantees, efficient rollout on such policies, and a novel upper bound on the optimal expected revenue.

Efficient Rollout of Static Policies: A static policy offers a (possibly random) assortment of resources without paying attention to the current state of the system. Letting n be the number of resources and T be the number of days in the booking horizon, to compute the value functions associated with a static policy, we need to solve a dynamic program with $O(2^{nT})$ possible states that keep track of the availability of each resource on each day. We show that we can compute the value functions of a static policy by using n separate dynamic programs, each with $O(T^2)$ possible states (Theorem 3.1). Intuitively speaking, this result is based on the following observation. Because each resource has unit capacity and customers use resources over consecutive days, if a resource is not available on day ℓ , then we cannot use this resource to serve a booking request for an interval

that starts on or before day $\ell - 1$ and ends on or after day $\ell + 1$. In our dynamic program, we keep track of the availability of each resource over uninterrupted intervals of days and use the fact that we cannot serve booking requests that straddle disjoint intervals. Thus, using our dynamic program, we can compute the value functions of a static policy in a number of operations that is polynomial in the number of resources and the number of days in the booking horizon.

Once we compute the value functions of a static policy, we can perform rollout on the static policy. Rolling out a static policy yields a policy that is guaranteed to perform at least as well as the static policy on hand; see Section 6.4.1 in Bertsekas (2017). Because the static policy does not consider the current state of the system, it may offer an unavailable resource in response to a booking request. We show that the rollout policy, in contrast, never offers an unavailable resource. The benefits from rolling out a static policy can be substantial. In our experiments, rolling out a static policy improves the performance of the static policy by up to 15%. In most rollout applications, the value functions of the static policy are approximated by using Monte Carlo simulation, but we can exactly compute the value functions of our static policies, precisely because each resource has unit capacity and customers use resources over consecutive days. Therefore, our results on computing the value functions of a static policy and performing rollout on a static policy both exploit the special structure of our revenue management problem.

Static Policies via Linear Approximations: By the discussion in the two previous paragraphs, if we have a static policy with a performance guarantee, then we can efficiently perform rollout on this static policy to obtain another policy that is at least as good. Letting D_{\max} be the maximum number of days of resource usage in a booking request, we use linear approximations of the value functions to give a static policy that is guaranteed to obtain $\frac{1}{2D_{\max}}$ fraction of the optimal total expected revenue (Theorem 4.1). Our result uses a characterization of feasible booking requests that holds under unit resource capacities. Earlier performance guarantees for linear value function approximations are in an asymptotic regime, where the resource capacities and the expected demand increase at the same rate; see, for example, Talluri and van Ryzin (1998). Our result does not require an asymptotic regime. Indeed, such an asymptotic regime is not relevant to us at all, because each resource is unique, so resource capacities are always one. Before our work, it was not known whether linear approximations could yield performance guarantees without asymptotic regimes.

Static Policies via Polynomial Approximations: Letting D_{\min} be the minimum number of days in a booking request, we use polynomial approximations of the value functions to give a static policy that is guaranteed to obtain $\frac{1}{2+\lceil(D_{\max}-1)/D_{\min}\rceil}$ fraction of the optimal total expected revenue. To establish this result, corresponding to each possible interval of days in a booking request, we designate a so-called intersection preserving subset of days with the following property. If we do

not have capacity on a day in the intersection preserving subset, then we can immediately conclude that we cannot accommodate a booking request for the corresponding interval. As a function of the number of days in the intersection preserving subsets, we give a performance guarantee for the static policy from the polynomial value function approximations (Theorem 6.1). Next, using the fact that resources have unit capacities and customers request resources over consecutive days, we bound the number of days in the intersection preserving subsets by $1 + \lceil (D_{\max} - 1)/D_{\min} \rceil$ (Theorem 6.2). Putting these two results together yields the desired performance guarantee. Surprisingly, if all booking requests are for the same duration ($D_{\max} = D_{\min}$), then our policy yields a constant $1/3$ performance guarantee. Our linear approximations have a looser guarantee than our polynomial ones, but the practical performance of both policies, especially after rollout, is competitive.

Upper Bound on the Optimal Policy Performance: We give an efficiently computable upper bound on the optimal total expected revenue. To assess the optimality gap of a policy, we can compare its total expected revenue with such an upper bound. Our upper bound is based on allocating the revenue from a booking request over different resources and solving a separate dynamic program to control the capacity of each resource. We show that this approach yields an upper bound on the optimal total expected revenue (Proposition 7.1). A common approach to obtain an upper bound is to formulate a linear programming approximation under the assumption that the choices of the customers take on their expected values; see Gallego et al. (2004). We show how to choose our revenue allocations such that the upper bound from our approach is at least as tight as that from such a linear program (Theorem 7.2). The dynamic program that we solve for each resource still has $O(2^T)$ states, keeping track of the availability of the resource on each day. We give an equivalent dynamic program with only $O(T^2)$ state variables (Theorem 7.3).

One motivation for using a linear program to construct an upper bound is that such upper bounds are tight in the asymptotic regime discussed earlier. This asymptotic regime does not apply to our problem because our resource capacities are always one, irrespective of the number of resources. In our experiments, the upper bounds from the linear program are substantially looser than our upper bounds, with gaps reaching 29%. Another approach to obtain an upper bound is based on decomposing the dynamic program for the problem by resources; see Topaloglu (2009). This approach constructs piecewise linear value function approximations that are separable by resources. In our problem, the capacity of each resource is one, so such a piecewise linear approximation is no different from a linear approximation. In our experiments, the upper bound from the decomposition approach was almost as poor as that from the linear program.

Validation on Real-World Hotel Data: We test our policies on a dataset from an actual boutique hotel, as well as randomly generated datasets. Our policies can handle real problem sizes. Policies

based on our linear and polynomial approximations outperform strong benchmarks. Moreover, rolling out a static policy can significantly improve the static policy on hand.

Literature Review: Rollout is a general approach to improving the performance of any policy; see Section 6.4.1 in Bertsekas (2017). The idea is to compute the value functions of the initial policy on hand and use the greedy policy with respect to the value functions. The policy from rollout is guaranteed to perform at least as well as the initial policy on hand. The difficulty of performing rollout is in computing the value functions of the initial policy. Computing these value functions can be as difficult as computing the optimal policy, so the value functions are often estimated by using simulation. We exploit the unit capacities of the resources and the interval structure of the booking requests to exactly compute the value functions of a static policy and to perform rollout on the static policy. The rollout approach has been used in combinatorial optimization, scheduling, vehicle routing, and revenue management, but most existing work uses approximations or simulations to estimate the value functions of the initial policy; see Bertsekas et al. (1997), Bertsekas and Castanon (1999), Secomandi (2001), and Bertsimas and Popescu (2003).

There is work on managing resources that possess reusable characteristics. In Manshadi and Rodilitz (2020), volunteers are asked to perform tasks, each volunteer becoming inactive for a random duration of time after she is tapped for a task. In Rusmevichientong et al. (2020), products are rented by customers for random durations of time and become available to be used by others once returned. In both papers, the authors give policies with performance guarantees. A critical differentiating factor of our problem is that a customer arriving into the system makes a booking request to use a resource on a future day. In other words, our problem setting involves advance reservations, which is, naturally, critical when working with lodging market places, hotels and freelancer matching platforms. In Manshadi and Rodilitz (2020), if a volunteer is tapped for a task, then her inactivity period immediately starts. In Rusmevichientong et al. (2020), if a customer arriving into the system rents a product, then the product is immediately checked out. As far as we are aware, it is not possible to incorporate advance reservations into these models.

For revenue management problems with non-unit resource capacities and booking requests not necessarily over intervals of days, letting L be the maximum number of resources used by a booking request, Baek and Ma (2019) and Ma et al. (2020) both give policies with a performance guarantee of $\frac{1}{1+L}$. The problem considered by these authors is more general than ours, but exploiting the special structure of our problem, we can improve the performance of their policies both theoretically and practically. In particular, because the authors consider non-unit resource capacities, it is not possible to perform rollout efficiently on their policies. In our experiments, rollout provides dramatic improvements in policy performance, reaching 15%. Also, the policies proposed by the authors,

when applied to our problem, provide a performance guarantee of $\frac{1}{1+D_{\max}}$. For $D_{\min} > 1$, we have $\frac{1}{2+\lceil(D_{\max}-1)/D_{\min}\rceil} > \frac{1}{1+D_{\max}}$, so if $D_{\min} > 1$, then our policy with the polynomial value function approximations has a better theoretical performance guarantee. The improvement provided by our policy is practically relevant because platforms may indeed impose minimum use requirements, so $D_{\min} > 1$. For example, in addition to its six rooms, Villa Mahal offers three villas: Ying Yang, Gunbatimi, and Ruya. These villas require a minimum stay of seven days. Our stronger theoretical performance guarantees translate into practical benefits. Our policies can improve the revenue performance of the policies in Baek and Ma (2019) and Ma et al. (2020) by 20%.

Some papers focus on building linear programming approximations for revenue management problems and characterize the optimality gaps of the policies derived from such approximations; see Talluri and van Ryzin (1998), Cooper (2002), and Jasin and Kumar (2012, 2013). These papers focus on the asymptotic regime discussed earlier in this section, but such a regime, once again, is not relevant to our setting because the capacities of our resources are always one. The policies derived from the linear programming approximations are often characterized by bid prices, where one attaches a bid price to each resource to capture the opportunity cost of a unit of resource. The decision for accepting a booking request is made by comparing the revenue from the booking request with the total opportunity cost of the resources used by the booking request; see Adelman (2007), Topaloglu (2009), Tong and Topaloglu (2013), Kirshner and Nediak (2015), Vossen and Zhang (2015a,b), and Kunnumkal and Talluri (2016a). The policies in this last set of papers do not have performance guarantees. There is work on incorporating customer choice into revenue management problems, where the customers choose among the offered booking options; see Gallego et al. (2004), Liu and van Ryzin (2008), Kunnumkal and Topaloglu (2008), Bront et al. (2009), and Meissner et al. (2012). Some papers focus on approximating the value functions under customer choice; see Zhang and Cooper (2005, 2009), Zhang and Adelman (2009), Kunnumkal and Topaloglu (2010), and Kunnumkal and Talluri (2016b). Others use stochastic approximation to compute booking limits and bid prices; see van Ryzin and Vulcano (2008), Topaloglu (2008), and Chaneton and Vulcano (2011). These papers do not give performance guarantees either.

Organization: In Section 2, we give a dynamic programming formulation for our problem. In Section 3, we show how to perform rollout efficiently on a static policy. In Section 4, we give a static policy using linear value function approximations. In Section 5, we establish a performance guarantee for this policy by showing that we can use our approximations to bound the performance of the optimal and static policy. In Section 6, we give a static policy using polynomial value function approximations. The performance guarantee for this policy also uses a bounding approach, but the specifics are different. In Section 7, we give a method to get an upper bound on the optimal total expected revenue. In Section 8, we give computational experiments. In Section 9, we conclude.

2. Problem Formulation

We have n unique resources indexed by $\mathcal{N} = \{1, \dots, n\}$. The resources are available for use during the days indexed by $\mathcal{T} = \{1, 2, \dots, T\}$. Let $\mathcal{F} = \{[s, f] : 1 \leq s \leq f \leq T\}$ denote the set of possible intervals of use, where the interval $[s, f]$ corresponds to a use over days $\{s, \dots, f\}$. The revenue from booking resource i over interval $[s, f]$ is $r_{i,[s,f]}$. The booking requests arrive over the time periods indexed by $\mathcal{Q} = \{1, 2, \dots, Q\}$. Each time period is a small enough interval of time that there is at most one booking request at each time period. At time period q , we have a booking request for using a resource over interval $[s, f]$ with probability $\lambda_{[s,f]}^q$. With probability $1 - \sum_{[s,f] \in \mathcal{F}} \lambda_{[s,f]}^q$, there is no booking request at time period q . Given that we offer assortment $S \subseteq \mathcal{N}$ of resources at time period q , the customer making a booking request at time period q chooses resource i with probability $\phi_i^q(S)$. The choice probability $\phi_i^q(S)$ is governed by a general choice model, as long as the choice probabilities of the resources in an assortment decrease as we add more resources into the assortment; that is, $\phi_i^q(S \cup \{j\}) \leq \phi_i^q(S)$ for all $i \in S$ and $j \notin S$.

Our goal is to find a policy for deciding which assortment of resources to make available for the customer arriving at each time period to maximize the total expected revenue from all booking requests. We formulate a dynamic program to compute an optimal policy. We use the vector $\mathbf{x} = (x_{i,\ell} : i \in \mathcal{N}, \ell \in \mathcal{T}) \in \{0, 1\}^{n \times T}$ to capture the state of the system at a generic time period, where we have $x_{i,\ell} = 1$ if and only if resource i is available for use on day ℓ . To accommodate a booking request to use resource i over the interval $[s, f]$, we need to have this resource available on days $\{s, \dots, f\}$. In other words, given that the state of the system is \mathbf{x} , we can accommodate a booking request to use resource i over the interval $[s, f]$ if and only if $\prod_{\ell=s}^f x_{i,\ell} = 1$. Let $J^q(\mathbf{x})$ be the optimal total expected revenue over time periods $\{q, \dots, Q\}$ given that the state of the system at time period q is \mathbf{x} . Using $\mathbf{e}_{i,[s,f]} \in \{0, 1\}^{n \times T}$ to denote the vector with ones only in the components corresponding to resource i and days $\{s, \dots, f\}$, we can find the optimal policy by computing the value functions $\{J^q : q \in \mathcal{Q}\}$ through the dynamic program

$$J^q(\mathbf{x}) = \sum_{[s,f] \in \mathcal{F}} \lambda_{[s,f]}^q \max_{S \subseteq \mathcal{N}} \left\{ \sum_{i \in \mathcal{N}} \phi_i^q(S) \left(\prod_{\ell=s}^f x_{i,\ell} \right) \left[r_{i,[s,f]} + J^{q+1}(\mathbf{x} - \mathbf{e}_{i,[s,f]}) \right] \right. \\ \left. + \left[\sum_{i \in \mathcal{N}} \phi_i^q(S) \left(1 - \prod_{\ell=s}^f x_{i,\ell} \right) + 1 - \sum_{i \in \mathcal{N}} \phi_i^q(S) \right] J^{q+1}(\mathbf{x}) \right\} + \left[1 - \sum_{[s,f] \in \mathcal{F}} \lambda_{[s,f]}^q \right] J^{q+1}(\mathbf{x}),$$

with the boundary condition that $J^{Q+1} = 0$. In this case, letting $\mathbf{e} \in \{0, 1\}^{n \times T}$ be the vector of all ones, the optimal total expected revenue is given by $J^1(\mathbf{e})$.

In the dynamic program above, if we offer the assortment S of resources to a customer with a resource request over the interval $[s, f]$, then she chooses resource i with probability $\phi_i^q(S)$. If

$\prod_{\ell=s}^f x_{i,\ell} = 1$, so that resource i is available to accommodate a booking over the interval $[s, f]$, then we generate a revenue of $r_{i,[s,f]}$ and resource i becomes unavailable over days $\{s, \dots, f\}$. With probability $\sum_{i \in \mathcal{N}} \phi_i^q(S) (1 - \prod_{\ell=s}^f x_{i,\ell})$, the customer chooses a resource that is not available on some day over the interval $[s, f]$. With probability $1 - \sum_{i \in \mathcal{N}} \phi_i^q(S)$, the customer does not choose any of the resources in the offered assortment. In either case, we do not consume capacity of any of the resources. Lastly, with probability $1 - \sum_{[s,f] \in \mathcal{F}} \lambda_{[s,f]}^q$, we do not have a booking request, in which case, we do not consume capacity of any of the resources either. In our dynamic program, we assume that if we have a customer making a booking request for the interval $[s, f]$, then we may offer a resource that is not available on one of the days $\{s, \dots, f\}$. If the customer chooses this resource, then she leaves without making a booking. This assumption is innocuous because, as we argue shortly, there exists an optimal policy that never offers an unavailable resource for a booking request. The dynamic program above can be unwieldy, but arranging the terms on the right side, we can write this dynamic program equivalently as

$$J^q(\mathbf{x}) = \sum_{[s,f] \in \mathcal{F}} \lambda_{[s,f]}^q \max_{S \subseteq \mathcal{N}} \left\{ \sum_{i \in \mathcal{N}} \phi_i^q(S) \left(\prod_{\ell=s}^f x_{i,\ell} \right) \left[r_{i,[s,f]} + J^{q+1}(\mathbf{x} - \mathbf{e}_{i,[s,f]}) - J^{q+1}(\mathbf{x}) \right] \right\} + J^{q+1}(\mathbf{x}). \quad (1)$$

Given that the state of the system at time period q is \mathbf{x} , we interpret $J^{q+1}(\mathbf{x}) - J^{q+1}(\mathbf{x} - \mathbf{e}_{i,[s,f]})$ as the opportunity cost of the capacities used by booking resource i for days $\{s, \dots, f\}$.

Using the dynamic program above, we can argue that there exists an optimal policy that never offers an unavailable resource for a booking request. In particular, given that the state of the system at time period q is \mathbf{x} and we have a booking request for the interval $[s, f]$, we can compute the optimal assortment of resources to offer by solving the maximization problem on the right side of (1). This maximization problem is of the form $\max_{S \subseteq \mathcal{N}} \sum_{i \in \mathcal{N}} \phi_i^q(S) p_{i,[s,f]}^q(\mathbf{x})$, where $p_{i,[s,f]}^q(\mathbf{x}) = (\prod_{\ell=s}^f x_{i,\ell}) [r_{i,[s,f]} + J^{q+1}(\mathbf{x} - \mathbf{e}_{i,[s,f]}) - J^{q+1}(\mathbf{x})]$. In Appendix A, we consider the problem $\max_{S \subseteq \mathcal{N}} \sum_{i \in \mathcal{N}} \phi_i^q(S) p_i$ and show that there exists an optimal solution S^* to this problem that satisfies $S^* \subseteq \{i \in \mathcal{N} : p_i > 0\}$. In other words, if $p_i^* \leq 0$, then S^* does not offer resource i . This result follows from the assumption that the choice probabilities satisfy $\phi_i^q(S \cup \{j\}) \leq \phi_i^q(S)$ for all $i \in S$ and $j \notin S$. If resource i is not available for some day over the interval $[s, f]$, then we have $\prod_{\ell=s}^f x_{i,\ell} = 0$, which implies that $p_{i,[s,f]}^q(\mathbf{x}) = 0$. Therefore, there exists an optimal solution to the maximization problem on the right side of (1) that does not offer resource i when this resource is unavailable for a booking request over the interval $[s, f]$.

The state variable \mathbf{x} in (1) has $O(2^{nT})$ possible values, making an optimal policy difficult to compute. Thus, we focus on developing policies with performance guarantees.

3. Efficient Rollout of Static Policies

In this section, we show that we can compute the value functions of a static policy efficiently, which ultimately allows us to perform rollout on the static policy. A static policy $\boldsymbol{\mu}$ is a collection of offer probabilities $\{\mu_{[s,f]}^q(S) : S \subseteq \mathcal{N}, [s,f] \in \mathcal{F}, q \in \mathcal{Q}\}$ such that if we have a booking request for interval $[s,f]$ at time period q , then the policy offers the assortment S of resources with probability $\mu_{[s,f]}^q(S)$. Naturally, we require $\sum_{S \subseteq \mathcal{N}} \mu_{[s,f]}^q(S) = 1$. Because the offer probabilities do not depend on the state of the system, a static policy may offer an unavailable resource. If a customer chooses an unavailable resource, then she leaves without making a booking. Nevertheless, we will ensure that, just like the optimal policy, the rollout policy based on a static policy never offers an unavailable resource. To compute the value functions of the static policy $\boldsymbol{\mu}$, we define $\psi_{\boldsymbol{\mu},i,[s,f]}^q = \sum_{S \subseteq \mathcal{N}} \mu_{[s,f]}^q(S) \phi_i^q(S)$, which is the probability that a customer arriving at time period q with a booking request for interval $[s,f]$ chooses resource i under the static policy $\boldsymbol{\mu}$. Let $J_{\boldsymbol{\mu}}^q(\mathbf{x})$ be the total expected revenue obtained by the static policy $\boldsymbol{\mu}$ over time periods $\{q, \dots, Q\}$ given that the state of the system at time period q is \mathbf{x} . We compute the value functions $\{J_{\boldsymbol{\mu}}^q : q \in \mathcal{Q}\}$ through the dynamic program

$$J_{\boldsymbol{\mu}}^q(\mathbf{x}) = \sum_{[s,f] \in \mathcal{F}} \lambda_{[s,f]}^q \left\{ \sum_{i \in \mathcal{N}} \psi_{\boldsymbol{\mu},i,[s,f]}^q \left(\prod_{\ell=s}^f x_{i,\ell} \right) \left[r_{i,[s,f]} + J_{\boldsymbol{\mu}}^{q+1}(\mathbf{x} - \mathbf{e}_{i,[s,f]}) - J_{\boldsymbol{\mu}}^{q+1}(\mathbf{x}) \right] \right\} + J_{\boldsymbol{\mu}}^{q+1}(\mathbf{x}), \quad (2)$$

with the boundary condition that $J_{\boldsymbol{\mu}}^{Q+1} = 0$. In this case, the total expected revenue obtained by the static policy $\boldsymbol{\mu}$ is given by $J_{\boldsymbol{\mu}}^1(\mathbf{e})$.

The dynamic program in (2) is similar to that in (1), but when computing the value functions of a static policy, the assortment that we offer is determined by the static policy, rather than being a decision variable. In particular, under the static policy $\boldsymbol{\mu}$, if we have a booking request for interval $[s,f]$ at time period q , then the customer chooses resource i with probability $\psi_{\boldsymbol{\mu},i,[s,f]}^q$. The state variable \mathbf{x} in (2) has $O(2^{nT})$ possible values, just like the state variable in (1), but two properties of static policies allow us to compute the value functions of a static policy efficiently. First, under the static policy $\boldsymbol{\mu}$, given that we have a booking request for interval $[s,f]$ at time period q , resource i receives a booking with fixed probability $\psi_{\boldsymbol{\mu},i,[s,f]}^q$. Therefore, intuitively speaking, each resource faces an exogenous stream of booking requests, allowing us to focus on each resource separately. Second, if resource i is available on days $\{a, \dots, b-1\}$ and days $\{b+1, \dots, c\}$, but not on day b , then this resource can never accommodate a booking request for an interval that starts before day $b-1$ and ends after day $b+1$. Thus, we can separately compute the total expected revenue from each uninterrupted interval of available days for a resource.

Motivated by the discussion above, to compute the value functions of the static policy $\boldsymbol{\mu}$, we let $V_{\boldsymbol{\mu},i}^q(a,b)$ be the total expected revenue collected by the static policy $\boldsymbol{\mu}$ from resource i over

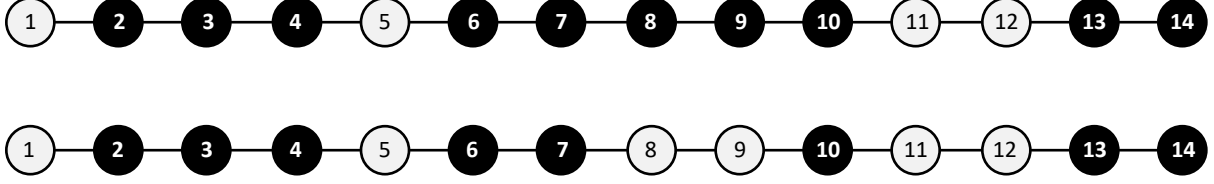


Figure 1 Maximal available intervals for \mathbf{x}_i (top) and maximal intervals for $\mathbf{x}_i - \mathbf{e}_{[8,9]}$ (bottom).

time periods $\{q, \dots, Q\}$ given that resource i is available for all the days in the interval $[a, b]$. We compute the value functions $\{V_{\mu,i}^q : q \in \mathcal{Q}\}$ by solving the dynamic program

$$V_{\mu,i}^q(a, b) = \sum_{[s,f] \subseteq [a,b]} \lambda_{[s,f]}^q \left\{ \psi_{\mu,i,[s,f]}^q \left[r_{i,[s,f]} + V_{\mu,i}^{q+1}(a, s-1) + V_{\mu,i}^{q+1}(f+1, b) - V_{\mu,i}^{q+1}(a, b) \right] \right\} + V_{\mu,i}^{q+1}(a, b), \quad (3)$$

with the boundary condition that $V_{\mu,i}^{Q+1} = 0$. Given that resource i is available on all days in the interval $[a, b]$, to compute the total expected revenue collected by the static policy μ from this resource over time periods $\{q, \dots, Q\}$, we consider the booking requests only for the intervals within the interval $[a, b]$. With probability $\lambda_{[s,f]}^q$, the customer arriving at time period q makes a booking request for interval $[s, f]$. Under the static policy μ , this customer chooses resource i with probability $\psi_{\mu,i,[s,f]}^q$, in which case we generate a revenue of $r_{i,[s,f]}$. Furthermore, resource i becomes no longer available on all days in the interval $[a, b]$, but it is still available for the intervals $[a, s-1]$ and $[f+1, b]$. Given that resource i is available on all days in the interval $[a, b]$ at time period q , we can interpret $V_{\mu,i}^{q+1}(a, b) - V_{\mu,i}^{q+1}(a, s-1) - V_{\mu,i}^{q+1}(f+1, b)$ in (3) as the opportunity cost of the capacities used by booking resource i for the days $\{s, \dots, f\}$. If $s = a$, then we set $V_{\mu,i}^{q+1}(a, s-1) = 0$, whereas if $f = b$, then we set $V_{\mu,i}^{q+1}(f+1, b) = 0$ in (3).

The state variable (a, b) in (3) has $O(T^2)$ possible values. Therefore, we can solve the dynamic program in (3) efficiently. Our main result in this section shows that we can compute the value functions $\{J_{\mu}^q : q \in \mathcal{Q}\}$ in (2) by using the value functions in $\{V_{\mu,i}^q : q \in \mathcal{Q}\}$ in (3). In particular, we use $\mathbf{x}_i = (x_{i,1}, \dots, x_{i,T}) \in \{0, 1\}^T$ to denote the state of resource i , where $x_{i,\ell} = 1$ if and only if resource i is available for use on day ℓ . We refer to the interval $[a, b]$ as a *maximal available interval* with respect to \mathbf{x}_i if and only if resource i is available on all days in the interval $[a, b]$, but not available on days $a-1$ and $b+1$; that is, $\prod_{\ell=a}^b x_{i,\ell} = 1$, $x_{i,a-1} = 0$ and $x_{i,b+1} = 0$. Let $\mathcal{I}(\mathbf{x}_i)$ be the collection of maximal available intervals with respect to \mathbf{x}_i . In the top portion of Figure 1, we show the collection $\mathcal{I}(\mathbf{x}_i)$ for a specific value of \mathbf{x}_i with $T = 14$. Each circle in the top portion corresponds to a day with $x_{i,\ell} = 1$ for the black circles and $x_{i,\ell} = 0$ for the white circles. The collection of maximal available intervals is $\mathcal{I}(\mathbf{x}_i) = \{[2, 4], [6, 10], [13, 14]\}$.

We use two properties of maximal available intervals. First, we have $\prod_{\ell=s}^f x_{i,\ell} = 1$ if and only if there exists a maximal available interval $[a, b] \in \mathcal{I}(\mathbf{x}_i)$ such that $[s, f] \subseteq [a, b]$. In the top portion

of Figure 1, for example, we have $\prod_{\ell=7}^9 x_{i,\ell} = 1$, so there must exist a maximal available interval $[a, b] \in \mathcal{I}(\mathbf{x}_i)$ such that $[7, 9] \subseteq [a, b]$. This maximal available interval is $[6, 10]$. Second, using $\mathbf{e}_{[s,f]} \in \{0, 1\}^T$ to denote the vector with ones only in the components corresponding to the days $\{s, \dots, f\}$, if $[a, b]$ is the maximal available interval that includes the interval $[s, f]$, then we have the identity $\mathcal{I}(\mathbf{x}_i - \mathbf{e}_{[s,f]}) = (\mathcal{I}(\mathbf{x}_i) \setminus [a, b]) \cup \{[a, s-1], [f+1, b]\}$. In this identity, if $s = a$, then we omit the interval $[a, s-1]$. Similarly, if $f = b$, then we omit the interval $[f+1, b]$. In the top portion of Figure 1, for example, we have $\mathcal{I}(\mathbf{x}_i) = \{[2, 4], [6, 10], [13, 14]\}$ for the specific value of \mathbf{x}_i . Considering the interval $[8, 9]$, the maximal available interval that includes this interval is $[6, 10]$. In the bottom portion of Figure 1, we have $\mathcal{I}(\mathbf{x}_i - \mathbf{e}_{[8,9]}) = \{[2, 4], [6, 7], [10, 10], [13, 14]\}$, which is indeed equal to $(\mathcal{I}(\mathbf{x}_i) \setminus [6, 10]) \cup \{[6, 7], [10, 10]\}$.

In the next theorem, we use these properties to show that we can compute the value functions $\{J_\mu^q : q \in \mathcal{Q}\}$ in (2) by using the value functions in $\{V_{\mu,i}^q : q \in \mathcal{Q}\}$ in (3)

Theorem 3.1 (Efficient Rollout) *If the value functions $\{J_\mu^q : q \in \mathcal{Q}\}$ are computed through (2) and the value functions $\{V_{\mu,i}^q : q \in \mathcal{Q}\}$ are computed through (3), then we have*

$$J_\mu^q(\mathbf{x}) = \sum_{i \in \mathcal{N}} \sum_{[a,b] \in \mathcal{I}(\mathbf{x}_i)} V_{\mu,i}^q(a, b).$$

Proof: We show the result by using induction over the time periods. At time period $Q+1$, we have $J_\mu^{Q+1} = 0 = V_{\mu,i}^{Q+1}$, so the result holds at time period $Q+1$. Assuming that the result holds at time period $q+1$, we show that the result holds at time period q as well. By the first property just before the theorem, we have $\prod_{\ell=s}^f x_{i,\ell} = 1$ if and only if there exists a maximal available interval $[a, b] \in \mathcal{I}(\mathbf{x}_i)$ such that $[s, f] \subseteq [a, b]$. Thus, using $\mathbb{1}_{\{\cdot\}}$ to denote the indicator function, we have $\prod_{\ell=s}^f x_{i,\ell} = 1$ if and only if $\sum_{[a,b] \in \mathcal{I}(\mathbf{x}_i)} \mathbb{1}_{\{[s,f] \subseteq [a,b]\}} = 1$. Furthermore, by the second property just before the theorem, if $[a, b]$ is the maximal available interval that includes the interval $[s, f]$, then we have $\mathcal{I}(\mathbf{x}_i - \mathbf{e}_{[s,f]}) = (\mathcal{I}(\mathbf{x}_i) \setminus [a, b]) \cup \{[a, s-1], [f+1, b]\}$. In this case, considering the maximal available interval $[a, b] \in \mathcal{I}(\mathbf{x}_i)$, for each interval $[s, f] \subseteq [a, b]$, using the induction hypothesis, we obtain the chain of equalities

$$\begin{aligned} J_\mu^{q+1}(\mathbf{x}) - J_\mu^{q+1}(\mathbf{x} - \mathbf{e}_{i,[s,f]}) &= \sum_{[c,d] \in \mathcal{I}(\mathbf{x}_i)} V_{\mu,i}^{q+1}(c, d) - \sum_{[c,d] \in \mathcal{I}(\mathbf{x}_i - \mathbf{e}_{i,[s,f]})} V_{\mu,i}^{q+1}(c, d) \\ &= \sum_{[c,d] \in \mathcal{I}(\mathbf{x}_i)} V_{\mu,i}^{q+1}(c, d) - \left(\sum_{[c,d] \in \mathcal{I}(\mathbf{x}_i)} V_{\mu,i}^{q+1}(c, d) - V_{\mu,i}^{q+1}(a, b) + V_{\mu,i}^{q+1}(a, s-1) + V_{\mu,i}^{q+1}(f+1, b) \right) \\ &= V_{\mu,i}^{q+1}(a, b) - V_{\mu,i}^{q+1}(a, s-1) - V_{\mu,i}^{q+1}(f+1, b), \end{aligned} \quad (4)$$

where the first equality holds because we have $J_\mu^{q+1}(\mathbf{x}) = \sum_{i \in \mathcal{N}} \sum_{[c,d] \in \mathcal{I}(\mathbf{x}_i)} V_{\mu,i}^{q+1}(c, d)$ by the induction hypothesis, whereas the second equality holds because, for all $[a, b] \in \mathcal{I}(\mathbf{x}_i)$ and

$[s, f] \subseteq [a, b]$, we have the identity $\mathcal{I}(\mathbf{x}_i - \mathbf{e}_{[s,f]}) = (\mathcal{I}(\mathbf{x}_i) \setminus [a, b]) \cup \{[a, s-1], [f+1, b]\}$. Thus, using the fact that $\prod_{\ell=s}^f x_{i,\ell} = 1$ if and only if $\sum_{[a,b] \in \mathcal{I}(\mathbf{x}_i)} \mathbb{1}_{\{[s,f] \subseteq [a,b]\}} = 1$, by (2), we obtain

$$\begin{aligned}
J_{\boldsymbol{\mu}}^q(\mathbf{x}) &= \sum_{[s,f] \in \mathcal{F}} \lambda_{[s,f]}^q \left\{ \sum_{i \in \mathcal{N}} \psi_{\boldsymbol{\mu},i,[s,f]}^q \left(\sum_{[a,b] \in \mathcal{I}(\mathbf{x}_i)} \mathbb{1}_{\{[s,f] \subseteq [a,b]\}} \right) \left[r_{i,[s,f]} + J_{\boldsymbol{\mu}}^{q+1}(\mathbf{x} - \mathbf{e}_{i,[s,f]}) - J_{\boldsymbol{\mu}}^{q+1}(\mathbf{x}) \right] \right\} + J_{\boldsymbol{\mu}}^{q+1}(\mathbf{x}) \\
&\stackrel{(a)}{=} \sum_{i \in \mathcal{N}} \sum_{[a,b] \in \mathcal{I}(\mathbf{x}_i)} \sum_{[s,f] \subseteq [a,b]} \lambda_{[s,f]}^q \left\{ \sum_{i \in \mathcal{N}} \psi_{\boldsymbol{\mu},i,[s,f]}^q \left[r_{i,[s,f]} + J_{\boldsymbol{\mu}}^{q+1}(\mathbf{x} - \mathbf{e}_{i,[s,f]}) - J_{\boldsymbol{\mu}}^{q+1}(\mathbf{x}) \right] \right\} + J_{\boldsymbol{\mu}}^{q+1}(\mathbf{x}) \\
&\stackrel{(b)}{=} \sum_{i \in \mathcal{N}} \sum_{[a,b] \in \mathcal{I}(\mathbf{x}_i)} \sum_{[s,f] \subseteq [a,b]} \lambda_{[s,f]}^q \left\{ \sum_{i \in \mathcal{N}} \psi_{\boldsymbol{\mu},i,[s,f]}^q \left[r_{i,[s,f]} - V_{\boldsymbol{\mu},i}^{q+1}(a,b) + V_{\boldsymbol{\mu},i}^{q+1}(a,s-1) + V_{\boldsymbol{\mu},i}^{q+1}(f+1,b) \right] \right\} \\
&\quad + \sum_{i \in \mathcal{N}} \sum_{[a,b] \in \mathcal{I}(\mathbf{x}_i)} V_{\boldsymbol{\mu},i}^{q+1}(a,b) \\
&\stackrel{(c)}{=} \sum_{i \in \mathcal{N}} \sum_{[a,b] \in \mathcal{I}(\mathbf{x}_i)} V_{\boldsymbol{\mu},i}^q(a,b),
\end{aligned}$$

where (a) follows by reordering the three sums on the left side of (a), (b) holds because (4) holds for all $[a,b] \in \mathcal{I}(\mathbf{x}_i)$ and $[s,f] \subseteq [a,b]$, along with the induction hypothesis, and (c) holds by (3). ■

The rollout policy from the static policy $\boldsymbol{\mu}$ makes its decisions by replacing $\{J^q : q \in \mathcal{Q}\}$ in (1) with $\{J_{\boldsymbol{\mu}}^q : q \in \mathcal{Q}\}$. We discuss the rollout policy from the static policy $\boldsymbol{\mu}$.

Rollout Policy Based on a Static Policy:

Given that the state of the system at time period q is \mathbf{x} and we have a booking request for interval $[s, f]$, the rollout policy from the static policy $\boldsymbol{\mu}$ offers the assortment of resources

$$S_{\boldsymbol{\mu},[s,f]}^{\text{Rollout},q}(\mathbf{x}) = \arg \max_{S \subseteq \mathcal{N}} \left\{ \sum_{i \in \mathcal{N}} \phi_i^q(S) \left(\prod_{\ell=s}^f x_{i,\ell} \right) \left[r_{i,[s,f]} + J_{\boldsymbol{\mu}}^{q+1}(\mathbf{x} - \mathbf{e}_{i,[s,f]}) - J_{\boldsymbol{\mu}}^{q+1}(\mathbf{x}) \right] \right\}. \quad (5)$$

The problem above is identical to the maximization problem on the right side of (1) after replacing $\{J^q : q \in \mathcal{Q}\}$ with $\{J_{\boldsymbol{\mu}}^q : q \in \mathcal{Q}\}$. By the same argument that we give just after (1), there exists an optimal solution $S_{\boldsymbol{\mu},[s,f]}^{\text{Rollout},q}(\mathbf{x})$ to the problem above such that $i \notin S_{\boldsymbol{\mu},[s,f]}^{\text{Rollout},q}(\mathbf{x})$ for each $i \in \mathcal{N}$ with $\prod_{\ell=s}^f x_{i,\ell} = 0$. Thus, the rollout policy never offers an unavailable resource for a booking request. It is a standard result that the total expected revenue obtained by the rollout policy from the static policy $\boldsymbol{\mu}$ is at least as large as the total expected revenue obtained by the static policy $\boldsymbol{\mu}$, so the rollout policy from the static policy $\boldsymbol{\mu}$ is guaranteed to be at least as good as the static policy $\boldsymbol{\mu}$; see Section 6.4.1 in Bertsekas (2017). Thus, if we have a performance guarantee for a static policy, then the same performance guarantee holds for the rollout policy based on this static policy as well. Our computational experiments indicate that the rollout policy based on the static policy $\boldsymbol{\mu}$ can perform significantly better than the static policy $\boldsymbol{\mu}$ itself.

Next, we give static policies with performance guarantees. By the discussion above, these performance guarantees hold for the corresponding rollout policies.

4. Resource Based Static Policy with Linear Approximations

We develop a static policy based on linear approximations of the value functions. In particular, we use linear value function approximations $\{\hat{J}_L^q : q \in \mathcal{Q}\}$ of the form

$$\hat{J}_L^q(\mathbf{x}) = \sum_{i \in \mathcal{N}} \sum_{\ell \in \mathcal{T}} \eta_{i,\ell}^q x_{i,\ell}.$$

In the approximation above, we interpret the coefficient $\eta_{i,\ell}^q$ as the opportunity cost of the capacity of resource i on day ℓ given that we are at time period q .

We use the following algorithm to compute the coefficients $\{\eta_{i,\ell}^q : i \in \mathcal{N}, \ell \in \mathcal{T}, q \in \mathcal{Q}\}$. For all $i \in \mathcal{N}$ and $t \in \mathcal{T}$, we set $\eta_{i,\ell}^{Q+1} = 0$. For each $q = Q, Q-1, \dots, 1$, we execute the two steps.

- **Construct Ideal Assortments:** For each $[s, f] \in \mathcal{F}$, compute the ideal assortment to offer to a customer with a booking request for interval $[s, f]$ as

$$A_{[s,f]}^q = \arg \max_{S \subseteq \mathcal{N}} \sum_{i \in \mathcal{N}} \phi_i^q(S) \left[r_{i,[s,f]} - \sum_{h=s}^f \eta_{i,h}^{q+1} \right]. \quad (6)$$

- **Compute Opportunity Costs:** For all $i \in \mathcal{N}$ and $\ell \in \mathcal{T}$, compute the opportunity cost of the capacity of resource i on day ℓ as

$$\eta_{i,\ell}^q = \sum_{[s,f] \in \mathcal{F}} \lambda_{[s,f]}^q \phi_i^q(A_{[s,f]}^q) \frac{\mathbb{1}_{\{\ell \in [s,f]\}}}{f-s+1} \left[r_{i,[s,f]} - \sum_{h=s}^f \eta_{i,h}^{q+1} \right] + \eta_{i,\ell}^{q+1}. \quad (7)$$

This algorithm fully specifies the parameters $\{\eta_{i,\ell}^q : i \in \mathcal{N}, \ell \in \mathcal{T}, q \in \mathcal{Q}\}$, which in turn specify our value function approximations. We give some intuition for the algorithm above.

In (6), $\sum_{h=s}^f \eta_{i,h}^{q+1}$ is the total opportunity cost of the capacities consumed by booking resource i for interval $[s, f]$ at time period q . Thus, $r_{i,[s,f]} - \sum_{h=s}^f \eta_{i,h}^{q+1}$ gives the revenue from booking resource i for interval $[s, f]$, net of the opportunity cost of the capacities consumed. In this case, $A_{[s,f]}^q$ is the assortment that maximizes the net expected revenue from a customer making a booking request for interval $[s, f]$ at time period q . In (7), we compute the opportunity costs of the capacity of resource i on day ℓ by using a recursion similar to that in (3). With probability $\lambda_{[s,f]}^q$, we have a customer arriving at time period q with a booking request for interval $[s, f]$. If we offer the ideal assortment $A_{[s,f]}^q$, then this customer chooses resource i with probability $\phi_i^q(A_{[s,f]}^q)$. At time period q , if we book resource i over interval $[s, f]$, then we obtain a net revenue of $r_{i,[s,f]} - \sum_{h=s}^f \eta_{i,h}^{q+1}$, but by doing so, we consume the capacity of resource i on each day $\ell \in \{s, \dots, f\}$. Noting the fraction $\frac{\mathbb{1}_{\{\ell \in [s,f]\}}}{f-s+1}$, we spread the net revenue evenly over each day $\ell \in \{s, \dots, f\}$.

The discussion in the previous paragraph provides only rough intuition for the algorithm we use to construct our linear value function approximations. Nevertheless, we will be able to use

this algorithm to develop a static policy with a performance guarantee that we can precisely quantify. In particular, noting the ideal assortments $\{A_{[s,f]}^q : [s,f] \in \mathcal{F}, q \in \mathcal{Q}\}$ computed through (6), we consider a static policy that always offers the assortment $A_{[s,f]}^q$ of resources to a customer making a booking request for interval $[s,f]$ at time period q . We refer to this static policy as the *resource based static policy* because our linear value function approximations have one component for each resource and day combination. In our problem context, the total number of resource and day combinations corresponds to the amount of resource capacity we have available. In the next theorem, we give a performance guarantee for the resource based static policy. In this theorem and throughout the rest of the paper, we let $D_{\max} = \max_{[s,f] \in \mathcal{F}} \{f - s + 1\}$, which corresponds to the maximum use duration for any possible booking request.

Theorem 4.1 (Performance of Resource Based Static Policy) *The resource based static policy obtains at least $1/(2D_{\max})$ fraction of the optimal total expected revenue.*

We give a proof for the theorem above in Section 5. The proof explicitly uses the fact that each resource is unique so that we have one unit of capacity for each resource on each day. When the resource capacities are not all one, it is unclear if we can use linear value function approximations to develop policies with performance guarantees. Being a static policy that offers each assortment of resources with fixed probabilities, the resource based static policy may end up offering an unavailable resource for a booking request, so it may not be appropriate to use this policy in practice. However, as discussed at the end of Section 3, the rollout policy from the resource based static policy never offers an unavailable resource. Furthermore, the rollout policy from the resource based static policy is guaranteed to perform at least as well as the resource based static policy, so the rollout policy inherits the performance guarantee of $1/(2D_{\max})$.

The coefficients $\{\eta_{i,\ell}^q : i \in \mathcal{N}, \ell \in \mathcal{T}, q \in \mathcal{Q}\}$ capture the opportunity costs of the capacities of different resources on different days, providing a natural interpretation of our linear value function approximations. Ma et al. (2020) use nonlinear value function approximations to give a policy with a performance guarantee of $1/(1 + D_{\max})$, which is stronger than the performance guarantee of $1/(2D_{\max})$ in Theorem 4.1. In our computational experiments, the practical performance of our linear value function approximations turns out to be comparable to that of nonlinear value function approximations. The competitive practical performance of our linear value function approximations, coupled with their interpretability, can make them particularly appealing. In Section 6, we will use nonlinear value function approximations to give another static policy that improves the performance guarantee in Ma et al. (2020) even further.

In the next section, we give a proof for Theorem 4.1.

5. Performance Guarantee for Resource Based Static Policy

The proof of Theorem 4.1 is based on giving an upper bound on the performance of the optimal policy and a lower bound on the performance of the resource based static policy.

5.1 Preliminary Lemmas

We will need three lemmas. Resource i is available for booking over interval $[s, f]$ if and only if $\prod_{\ell=s}^f x_{i,\ell} = 1$. In the next lemma, we give a lower bound on $\prod_{\ell=s}^f x_{i,\ell}$ that is linear in \mathbf{x} .

Lemma 5.1 (Linear Proxy for Resource Availability Condition) *For each $\mathbf{x} \in \{0, 1\}^{n \times T}$, $i \in \mathcal{N}$, and $[s, f] \in \mathcal{F}$, we have*

$$\prod_{\ell=s}^f x_{i,\ell} \geq \frac{(\sum_{\ell=s}^f x_{i,\ell}) - (f - s)}{1 + f - s}.$$

Proof: First, assume that $\prod_{\ell=s}^f x_{i,\ell} = 1$. Thus, we have $x_{i,\ell} = 1$ for all $\ell = s, \dots, f$, so that $\sum_{\ell=s}^f x_{i,\ell} = f - s + 1$. In this case, the left side of the inequality in the lemma is one, whereas the right side is $\frac{1}{1+f-s}$. Because $1 \geq \frac{1}{1+f-s}$, the inequality holds whenever $\prod_{\ell=s}^f x_{i,\ell} = 1$. Second, assume that $\prod_{\ell=s}^f x_{i,\ell} = 0$. Thus, we have $x_{i,\ell} = 0$ for some $\ell = s, \dots, f$, so that $\sum_{\ell=s}^f x_{i,\ell} \leq f - s$. In this case, noting that $\sum_{\ell=s}^f x_{i,\ell} - (f - s) \leq 0$, the left side of the inequality in the lemma is zero, whereas the right side is at most zero, so the inequality holds whenever $\prod_{\ell=s}^f x_{i,\ell} = 0$. ■

The above lemma requires every resource to be unique so that $\mathbf{x} \in \{0, 1\}^{n \times T}$. The next lemma shows that the contribution of each resource to the objective value in (6) is nonnegative.

Lemma 5.2 (Nonnegative Contribution of Each Resource) *For each $i \in \mathcal{N}$, $[s, f] \in \mathcal{F}$, and $q \in \mathcal{Q}$, we have $\phi_i^q(A_{[s,f]}^q)[r_{i,[s,f]} - \sum_{h=s}^f \eta_{i,h}^{q+1}] \geq 0$.*

Proof: For notational brevity, fixing $[s, f] \in \mathcal{F}$ and $q \in \mathcal{Q}$, let $p_i = r_{i,[s,f]} - \sum_{h=s}^f \eta_{i,h}^{q+1}$ for each $i \in \mathcal{N}$. Suppose on the contrary that we have $\phi_k^q(A_{[s,f]}^q) p_k < 0$ for some $k \in \mathcal{N}$. Then, we have $p_k < 0$ and $\phi_k^q(A_{[s,f]}^q) > 0$. Noting $\phi_k^q(A_{[s,f]}^q) > 0$, since the booking probability of a resource that is not offered is zero, it must be the case that $k \in A_{[s,f]}^q$. We partition the assortment of resources $A_{[s,f]}^q$ into $A^+ = \{j \in A_{[s,f]}^q : p_j \geq 0\}$ and $A^- = \{j \in A_{[s,f]}^q : p_j < 0\}$. Using the fact that we have $k \in A^-$ and $\phi_k^q(A_{[s,f]}^q) p_k < 0$, we have the chain of inequalities

$$\sum_{i \in \mathcal{N}} \phi_i^q(A_{[s,f]}^q) p_i = \sum_{i \in A^+} \phi_i^q(A_{[s,f]}^q) p_i + \sum_{i \in A^-} \phi_i^q(A_{[s,f]}^q) p_i < \sum_{i \in A^+} \phi_i^q(A_{[s,f]}^q) p_i \stackrel{(a)}{\leq} \sum_{i \in A^+} \phi_i^q(A^+) p_i,$$

where (a) uses the assumption that $\phi_i^q(S \cup \{j\}) \leq \phi_i^q(S)$ for all $i \in S$ and $j \notin S$. The chain of inequalities above contradicts the fact that $A_{[s,f]}^q$ is an optimal solution to problem (6). ■

In the next lemma, we give an inequality that will become useful to lower bound the total expected revenue obtained by the resource based static policy.

Lemma 5.3 (Upper Bound on Net Expected Revenue) *Letting $\{A_{[s,f]}^q : [s,f] \in \mathcal{F}, q \in \mathcal{Q}\}$ and $\{\eta_{i,\ell}^q : i \in \mathcal{N}, \ell \in \mathcal{T}, q \in \mathcal{Q}\}$ be computed through (6) and (7), we have*

$$\sum_{[s,f] \in \mathcal{F}} \lambda_{[s,f]}^q \sum_{i \in \mathcal{N}} \phi_i^q(A_{[s,f]}^q) \frac{f-s}{f-s+1} \left[r_{i,[s,f]} - \sum_{h=s}^f \eta_{i,h}^{q+1} \right] \leq \frac{D_{\max} - 1}{D_{\max}} (\hat{J}_{\mathcal{L}}^q(\mathbf{e}) - \hat{J}_{\mathcal{L}}^{q+1}(\mathbf{e})).$$

Proof: Because $x/(x+1)$ is increasing in $x \in \mathbb{R}_+$ and $f-s+1 \leq D_{\max}$, we have $\frac{f-s}{f-s+1} \leq \frac{D_{\max}-1}{D_{\max}}$. Noting that $\phi_i^q(A_{[s,f]}^q) [r_{i,[s,f]} - \sum_{h=s}^f \eta_{i,h}^{q+1}] \geq 0$ for all $i \in \mathcal{N}$ and $[s,f] \in \mathcal{F}$ by Lemma 5.2, we get

$$\begin{aligned} & \sum_{[s,f] \in \mathcal{F}} \lambda_{[s,f]}^q \sum_{i \in \mathcal{N}} \phi_i^q(A_{[s,f]}^q) \frac{f-s}{f-s+1} \left[r_{i,[s,f]} - \sum_{h=s}^f \eta_{i,h}^{q+1} \right] \\ & \leq \frac{D_{\max} - 1}{D_{\max}} \sum_{i \in \mathcal{N}} \sum_{[s,f] \in \mathcal{F}} \lambda_{[s,f]}^q \phi_i^q(A_{[s,f]}^q) \left[r_{i,[s,f]} - \sum_{h=s}^f \eta_{i,h}^{q+1} \right] \\ & \stackrel{(a)}{=} \frac{D_{\max} - 1}{D_{\max}} \sum_{i \in \mathcal{N}} \sum_{\ell \in \mathcal{T}} \sum_{[s,f] \in \mathcal{F}} \lambda_{[s,f]}^q \phi_i^q(A_{[s,f]}^q) \frac{\mathbb{1}_{\{\ell \in [s,f]\}}}{f-s+1} \left[r_{i,[s,f]} - \sum_{h=s}^f \eta_{i,h}^{q+1} \right] \\ & \stackrel{(b)}{=} \frac{D_{\max} - 1}{D_{\max}} \sum_{i \in \mathcal{N}} \sum_{\ell \in \mathcal{T}} (\eta_{i,\ell}^q - \eta_{i,\ell}^{q+1}) \stackrel{(c)}{=} \frac{D_{\max} - 1}{D_{\max}} (\hat{J}_{\mathcal{L}}^q(\mathbf{e}) - \hat{J}_{\mathcal{L}}^{q+1}(\mathbf{e})), \end{aligned}$$

where (a) holds by the identity $\sum_{\ell \in \mathcal{T}} \frac{\mathbb{1}_{\{\ell \in [s,f]\}}}{f-s+1} = 1$, (b) follows by (7), and (c) follows because we have $\hat{J}_{\mathcal{L}}^q(\mathbf{x}) = \sum_{i \in \mathcal{N}} \sum_{\ell \in \mathcal{T}} \eta_{i,\ell}^q x_{i,\ell}$.

We focus on the first part of the proof of Theorem 4.1, which constructs an upper bound on the optimal total expected revenue.

5.2 Upper Bound on the Optimal Total Expected Revenue

Letting the value functions $\{J^q : q \in \mathcal{Q}\}$ be computed through (1), it is a standard result that if, for all $\mathbf{x} \in \{0,1\}^{n \times T}$ and $q \in \mathcal{Q}$, the value function approximations $\{\tilde{J}^q : q \in \mathcal{Q}\}$ satisfy

$$\tilde{J}^q(\mathbf{x}) \geq \sum_{[s,f] \in \mathcal{F}} \lambda_{[s,f]}^q \max_{S \subseteq \mathcal{N}} \left\{ \sum_{i \in \mathcal{N}} \phi_i^q(S) \left(\prod_{\ell=s}^f x_{i,\ell} \right) \left[r_{i,[s,f]} + \tilde{J}^{q+1}(\mathbf{x} - \mathbf{e}_{i,[s,f]}) - \tilde{J}^{q+1}(\mathbf{x}) \right] \right\} + \tilde{J}^{q+1}(\mathbf{x}), \quad (8)$$

then we have $\tilde{J}^q(\mathbf{x}) \geq J^q(\mathbf{x})$ for all $\mathbf{x} \in \{0,1\}^{n \times T}$ and $q \in \mathcal{Q}$; see Section 5.3.3 in Bertsekas (2017). This result is known as the monotonicity of the dynamic programming operator. Note that the inequality above is the version of (1) in the greater than or equal to sense. Thus, if the value function approximations $\{\tilde{J}_q : q \in \mathcal{Q}\}$ satisfy (1) in the greater than or equal to sense, then they form an upper bound on the optimal value functions $\{J^q : q \in \mathcal{Q}\}$ that satisfy (1) in the equality sense. In the next proposition, we use this result to show that $2\hat{J}_{\mathcal{L}}^1(\mathbf{e})$ provides an upper bound on the optimal total expected revenue. Thus, we can use our linear value function approximations to come up with an upper bound on the optimal total expected revenue.

Proposition 5.4 (Upper Bound on Optimal Performance) *Noting that the optimal total expected revenue is $J^1(\mathbf{e})$, we have $J^1(\mathbf{e}) \leq 2\hat{J}_L^1(\mathbf{e})$.*

Proof: Using our linear value function approximations $\{\hat{J}_L^q : q \in \mathcal{Q}\}$ computed through (6) and (7), let $\tilde{J}^q(\mathbf{x}) = \hat{J}_L^q(\mathbf{e}) + \hat{J}_L^q(\mathbf{x})$. We show that $\{\tilde{J}^q : q \in \mathcal{Q}\}$ satisfies (8). In particular, we have

$$\begin{aligned}
 & \sum_{[s,f] \in \mathcal{F}} \lambda_{[s,f]}^q \max_{S \subseteq \mathcal{N}} \left\{ \sum_{i \in \mathcal{N}} \phi_i^q(S) \left(\prod_{\ell=s}^f x_{i,\ell} \right) \left[r_{i,[s,f]} + \tilde{J}^{q+1}(\mathbf{x}) - \tilde{J}^{q+1}(\mathbf{x} - \mathbf{e}_{i,[s,f]}) \right] \right\} \\
 &= \sum_{[s,f] \in \mathcal{F}} \lambda_{[s,f]}^q \max_{S \subseteq \mathcal{N}} \left\{ \sum_{i \in \mathcal{N}} \phi_i^q(S) \left(\prod_{\ell=s}^f x_{i,\ell} \right) \left[r_{i,[s,f]} - \sum_{h=s}^f \eta_{i,h}^{q+1} \right] \right\} \\
 &\stackrel{(a)}{\leq} \sum_{[s,f] \in \mathcal{F}} \lambda_{[s,f]}^q \max_{S \subseteq \mathcal{N}} \left\{ \sum_{i \in \mathcal{N}} \phi_i^q(S) \left[r_{i,[s,f]} - \sum_{h=s}^f \eta_{i,h}^{q+1} \right] \right\} \\
 &\stackrel{(b)}{=} \sum_{[s,f] \in \mathcal{F}} \lambda_{[s,f]}^q \sum_{i \in \mathcal{N}} \phi_i^q(A_{[s,f]}^q) \left[r_{i,[s,f]} - \sum_{h=s}^f \eta_{i,h}^{q+1} \right] \\
 &\stackrel{(c)}{=} \sum_{i \in \mathcal{N}} \sum_{[s,f] \in \mathcal{F}} \lambda_{[s,f]}^q \phi_i^q(A_{[s,f]}^q) \frac{\sum_{\ell \in \mathcal{T}} \mathbb{1}_{\{\ell \in [s,f]\}}}{f-s+1} \left[r_{i,[s,f]} - \sum_{h=s}^f \eta_{i,h}^{q+1} \right] \\
 &= \sum_{i \in \mathcal{N}} \sum_{\ell \in \mathcal{T}} \sum_{[s,f] \in \mathcal{F}} \lambda_{[s,f]}^q \phi_i^q(A_{[s,f]}^q) \frac{\mathbb{1}_{\{\ell \in [s,f]\}}}{f-s+1} \left[r_{i,[s,f]} - \sum_{h=s}^f \eta_{i,h}^{q+1} \right] \\
 &\stackrel{(d)}{=} \sum_{i \in \mathcal{N}} \sum_{\ell \in \mathcal{T}} (\eta_{i,\ell}^q - \eta_{i,\ell}^{q+1}) \stackrel{(e)}{\leq} \sum_{i \in \mathcal{N}} \sum_{\ell \in \mathcal{T}} (\eta_{i,\ell}^q - \eta_{i,\ell}^{q+1}) (1 + x_{i,\ell}) = \tilde{J}^q(\mathbf{x}) - \tilde{J}^{q+1}(\mathbf{x}).
 \end{aligned}$$

The inequality (a) holds because Lemma A.1 in Appendix A implies that if $r_{i,[s,f]} - \sum_{h=s}^f \eta_{i,h}^{q+1} \leq 0$, then there exists an optimal solution to the problem on the left side of (a) such that resource i is not offered. So, if $r_{i,[s,f]} - \sum_{h=s}^f \eta_{i,h}^{q+1} \leq 0$, then we can drop resource i from consideration in this problem, but if $r_{i,[s,f]} - \sum_{h=s}^f \eta_{i,h}^{q+1} > 0$, then we have $(\prod_{\ell=s}^f x_{i,\ell}) [r_{i,[s,f]} - \sum_{h=s}^f \eta_{i,h}^{q+1}] \leq r_{i,[s,f]} - \sum_{h=s}^f \eta_{i,h}^{q+1}$. Moreover, (b) follows from the definition of $A_{[s,f]}^q$ in (6), (c) uses the fact that $\sum_{\ell \in \mathcal{T}} \mathbb{1}_{\{\ell \in [s,f]\}} = f - s + 1$, and (d) follows from the definition of $\eta_{i,\ell}^q$ in (7). Lastly, (e) holds because we have $\phi_i^q(A_{[s,f]}^q) [r_{i,[s,f]} - \sum_{h=s}^f \eta_{i,h}^{q+1}] \geq 0$ by Lemma 5.2, which implies that $\eta_{i,\ell}^q \geq \eta_{i,\ell}^{q+1}$ by (7).

The chain of inequalities above shows that $\{\tilde{J}^q : q \in \mathcal{Q}\}$ satisfies (8). Thus, we have $\tilde{J}^q(\mathbf{x}) \geq J^q(\mathbf{x})$ for all $\mathbf{x} \in \{0, 1\}^{n \times T}$ and $q \in \mathcal{Q}$, so $J^1(\mathbf{e}) \leq \tilde{J}^1(\mathbf{e}) = 2\hat{J}_L^1(\mathbf{e})$. \blacksquare

We focus on the second part of the proof of Theorem 4.1, which constructs a lower bound on the total expected revenue of the resource based static policy.

5.3 Lower Bound on the Performance of the Static Policy

We give a dynamic program to compute the total expected revenue of the resource based static policy. Let $U_L^q(\mathbf{x})$ be the total expected revenue obtained by the resource based static policy over

time periods $\{q, \dots, Q\}$ given that the state of the system at time period q is \mathbf{x} . We can compute the value functions $\{U_L^q : q \in \mathcal{Q}\}$ through the dynamic program

$$U_L^q(\mathbf{x}) = \sum_{[s,f] \in \mathcal{F}} \lambda_{[s,f]}^q \left\{ \sum_{i \in \mathcal{N}} \phi_i^q(A_{[s,f]}^q) \left(\prod_{\ell=s}^f x_{i,\ell} \right) \left[r_{i,[s,f]} + U_L^{q+1}(\mathbf{x} - \mathbf{e}_{i,[s,f]}) - U_L^{q+1}(\mathbf{x}) \right] \right\} + U_L^{q+1}(\mathbf{x}), \quad (9)$$

with the boundary condition that $U_L^{Q+1} = 0$. The dynamic program above is similar to that in (2), but under the resource based static policy, a customer making a booking request for interval $[s, f]$ at time period q chooses resource i with probability $\phi_i^q(A_{[s,f]}^q)$. In the next proposition, we use the linear value function approximations $\{\hat{J}_L^q : q \in \mathcal{Q}\}$ to give a lower bound on the performance of the resource based static policy. Lemma 5.1 plays an important role in the proof of the next proposition. Thus, the lower bound on the performance of the resource based static policy explicitly uses the fact that the resources are unique so that each has a capacity of one.

Proposition 5.5 (Lower Bound on Static Policy Performance) *Letting the value functions $\{U_L^q : q \in \mathcal{Q}\}$ be computed through (9), for each $\mathbf{x} \in \{0, 1\}^{n \times T}$ and $q \in \mathcal{Q}$, we have*

$$U_L^q(\mathbf{x}) \geq \hat{J}_L^q(\mathbf{x}) - \frac{D_{\max} - 1}{D_{\max}} \hat{J}_L^q(\mathbf{e}).$$

Proof: We give an inequality that will be useful later in the proof. Because $\prod_{\ell=s}^f x_{i,\ell} \geq \frac{\sum_{\ell=s}^f x_{i,\ell} - (f-s)}{1+f-s}$ by Lemma 5.1 and $\phi_i^q(A_{[s,f]}^q) [r_{i,[s,f]} - \sum_{h=s}^f \eta_{i,h}^{q+1}] \geq 0$ by Lemma 5.2, we have

$$\begin{aligned} & \sum_{[s,f] \in \mathcal{F}} \lambda_{[s,f]}^q \sum_{i \in \mathcal{N}} \phi_i^q(A_{[s,f]}^q) \left(\prod_{\ell=s}^f x_{i,\ell} \right) \left[r_{i,[s,f]} + \hat{J}_L^{q+1}(\mathbf{x} - \mathbf{e}_{i,[s,f]}) - \hat{J}_L^{q+1}(\mathbf{x}) \right] \\ &= \sum_{[s,f] \in \mathcal{F}} \lambda_{[s,f]}^q \sum_{i \in \mathcal{N}} \phi_i^q(A_{[s,f]}^q) \left(\prod_{\ell=s}^f x_{i,\ell} \right) \left[r_{i,[s,f]} - \sum_{h=s}^f \eta_{i,h}^{q+1} \right] \\ &\geq \sum_{[s,f] \in \mathcal{F}} \lambda_{[s,f]}^q \sum_{i \in \mathcal{N}} \phi_i^q(A_{[s,f]}^q) \frac{\sum_{\ell \in \mathcal{T}} \mathbb{1}_{\{\ell \in [s,f]\}} x_{i,\ell} - (f-s)}{f-s+1} \left[r_{i,[s,f]} - \sum_{h=s}^f \eta_{i,h}^{q+1} \right] \\ &\stackrel{(a)}{\geq} \sum_{i \in \mathcal{N}} \sum_{\ell \in \mathcal{T}} x_{i,\ell} \left\{ \sum_{[s,f] \in \mathcal{F}} \lambda_{[s,f]}^q \phi_i^q(A_{[s,f]}^q) \frac{\mathbb{1}_{\{\ell \in [s,f]\}}}{f-s+1} \left[r_{i,[s,f]} - \sum_{h=s}^f \eta_{i,h}^{q+1} \right] \right\} - \frac{D_{\max} - 1}{D_{\max}} (\hat{J}_L^q(\mathbf{e}) - \hat{J}_L^{q+1}(\mathbf{e})) \\ &\stackrel{(b)}{=} \sum_{i \in \mathcal{N}} \sum_{\ell \in \mathcal{T}} x_{i,\ell} (\eta_{i,\ell}^q - \eta_{i,\ell}^{q+1}) - \frac{D_{\max} - 1}{D_{\max}} (\hat{J}_L^q(\mathbf{e}) - \hat{J}_L^{q+1}(\mathbf{e})) \\ &\stackrel{(c)}{=} \hat{J}_L^q(\mathbf{x}) - \hat{J}_L^{q+1}(\mathbf{x}) - \frac{D_{\max} - 1}{D_{\max}} (\hat{J}_L^q(\mathbf{e}) - \hat{J}_L^{q+1}(\mathbf{e})), \end{aligned}$$

where (a) follows by arranging the terms on the left side of (a) and using Lemma 5.3, (b) uses (7), and (c) uses the fact that $\hat{J}_L^q(\mathbf{x}) = \sum_{i \in \mathcal{N}} \sum_{\ell \in \mathcal{T}} \eta_{i,\ell} x_{i,\ell}$.

In the rest of the proof, we show the inequality in the proposition by using induction over the time periods. At time period $Q+1$, because we have $U_L^{Q+1} = 0 = \hat{J}_L^{Q+1}$, the inequality holds at time

period $Q + 1$. Assuming that the inequality holds at time period $q + 1$, we show that the inequality holds at time period q as well. Arranging the terms, the coefficient of $U_{\mathcal{L}}^{q+1}(\mathbf{x})$ on the right side of (9) is $1 - \sum_{[s,f] \in \mathcal{F}} \lambda_{[s,f]}^q \sum_{i \in \mathcal{N}} \phi_i^q(A_{[s,f]}^q) \prod_{\ell=s}^f x_{i,\ell}$. Because $\sum_{[s,f] \in \mathcal{F}} \lambda_{[s,f]}^q \leq 1$ and $\sum_{i \in \mathcal{N}} \phi_i^q(A_{[s,f]}^q) \leq 1$, this coefficient is nonnegative. Letting $\alpha^q = \frac{D_{\max}-1}{D_{\max}} \hat{J}_{\mathcal{L}}^q(\mathbf{e})$ for notational brevity. By the induction assumption, we have $U_{\mathcal{L}}^{q+1}(\mathbf{x}) \geq \hat{J}_{\mathcal{L}}^{q+1}(\mathbf{x}) - \alpha^{q+1}$ for all $\mathbf{x} \in \{0, 1\}^{n \times T}$. In this case, replacing $U_{\mathcal{L}}^{q+1}(\mathbf{x})$ and $U_{\mathcal{L}}^{q+1}(\mathbf{x} - \mathbf{e}_{i,[s,f]})$ on the right side of (9) with their corresponding lower bounds $\hat{J}_{\mathcal{L}}^{q+1}(\mathbf{x}) - \alpha^{q+1}$ and $\hat{J}_{\mathcal{L}}^{q+1}(\mathbf{x} - \mathbf{e}_{i,[s,f]}) - \alpha^{q+1}$, respectively, the right side of (9) gets smaller, so we get

$$\begin{aligned} U_{\mathcal{L}}^q(\mathbf{x}) &\geq \sum_{[s,f] \in \mathcal{F}} \lambda_{[s,f]}^q \left\{ \sum_{i \in \mathcal{N}} \phi_i^q(A_{[s,f]}^q) \left(\prod_{\ell=s}^f x_{i,\ell} \right) \left[r_{i,[s,f]} + \hat{J}_{\mathcal{L}}^{q+1}(\mathbf{x} - \mathbf{e}_{i,[s,f]}) - \hat{J}_{\mathcal{L}}^{q+1}(\mathbf{x}) \right] \right\} + \hat{J}_{\mathcal{L}}^{q+1}(\mathbf{x}) - \alpha^{q+1} \\ &\stackrel{(d)}{\geq} \hat{J}_{\mathcal{L}}^q(\mathbf{x}) - \hat{J}_{\mathcal{L}}^{q+1}(\mathbf{x}) - (\alpha^q - \alpha^{q+1}) + \hat{J}_{\mathcal{L}}^{q+1}(\mathbf{x}) - \alpha^{q+1} = \hat{J}_{\mathcal{L}}^q(\mathbf{x}) - \alpha^q, \end{aligned}$$

where (d) uses the chain of inequalities earlier in the proof, along with $\frac{D_{\max}-1}{D_{\max}} (\hat{J}_{\mathcal{L}}^q(\mathbf{e}) - \hat{J}_{\mathcal{L}}^{q+1}(\mathbf{e})) = \alpha^q - \alpha^{q+1}$. Thus, the inequality in the proposition holds at time period q . \blacksquare

Finally, we can use Propositions 5.4 and 5.5 to give a proof for Theorem 4.1.

Proof of Theorem 4.1:

By Proposition 5.4, we have $\hat{J}_{\mathcal{L}}^1(\mathbf{e}) \geq \frac{1}{2} J^1(\mathbf{e})$. Using Proposition 5.5 with $\mathbf{x} = \mathbf{e}$ and $q = 1$, we have $U_{\mathcal{L}}^1(\mathbf{e}) \geq \hat{J}_{\mathcal{L}}^1(\mathbf{e}) - \frac{D_{\max}-1}{D_{\max}} \hat{J}_{\mathcal{L}}^1(\mathbf{e}) = \frac{1}{D_{\max}} \hat{J}_{\mathcal{L}}^1(\mathbf{e})$, so we get $U_{\mathcal{L}}^1(\mathbf{e}) \geq \frac{1}{D_{\max}} \hat{J}_{\mathcal{L}}^1(\mathbf{e}) \geq \frac{1}{2D_{\max}} J^1(\mathbf{e})$. \blacksquare

When we compute the ideal assortments $\{A_{[s,f]}^q : [s,f] \in \mathcal{F}, q \in \mathcal{Q}\}$ through (6) and (7), we need to solve a problem of the form $\max_{S \subseteq \mathcal{N}} \sum_{i \in \mathcal{N}} \phi_i^q(S) p_i$ for fixed $\{p_i : i \in \mathcal{N}\}$. Viewing p_i as the net revenue contribution from booking resource i , this problem finds an assortment of resources to offer to a customer to maximize the expected net revenue contribution. Such an assortment optimization problem is of combinatorial nature, but there are a variety of choice models characterizing the choice probabilities $\{\phi_i^q(S) : i \in S, S \subseteq \mathcal{N}, q \in \mathcal{Q}\}$, under which we can give efficient algorithms to solve the assortment optimization problem. For example, there are polynomial time algorithms to solve this problem under the multinomial logit, mixture of independent and multinomial logit, nested logit, preference list, and Markov chain choice models; see Talluri and van Ryzin (2004), Davis et al. (2014), Blanchet et al. (2016), Aouad et al. (2020), and Cao et al. (2020). Using **Assort** to denote the number of operations needed to solve the assortment optimization problem and noting that $O(|\mathcal{F}|) = O(D_{\max} T)$, we can compute the ideal assortments through (6)-(7) in $O(\text{Assort } D_{\max} T Q + n D_{\max}^2 T Q)$ operations, because in (7), the number of intervals containing $\ell \in \mathcal{T}$ is $O(D_{\max}^2)$. Once we compute the ideal assortments, we can compute the value functions $\{J_{\mu}^q : q \in \mathcal{Q}\}$ of the rollout policy through (3) in $O(n D_{\max} T^3 Q)$ operations.

Next, we give a static policy using nonlinear value function approximations.

6. Itinerary Based Static Policy with Polynomial Approximations

We develop a static policy based on polynomial approximations of the value functions. In particular, we use polynomial value function approximations $\{\hat{J}_p^q : q \in \mathcal{Q}\}$ of the form

$$\hat{J}_p^q(\mathbf{x}) = \sum_{i \in \mathcal{N}} \sum_{[s,f] \in \mathcal{F}} \gamma_{i,[s,f]}^q \prod_{\ell=s}^f x_{i,\ell}.$$

Here, we have one component for each resource and interval pair. The component for resource i and interval $[s, f]$ is a function of the availability of resource i over interval $[s, f]$. To choose the coefficients $\{\gamma_{i,[s,f]}^q : i \in \mathcal{N}, [s, f] \in \mathcal{F}, q \in \mathcal{Q}\}$, we will use a succinct representation of whether two intervals overlap. We refer to the subset of days $C_{[s,f]} \subseteq \mathcal{T}$ as the *intersection preserving subset* for the interval $[s, f]$ if it satisfies the following two properties. (i) We have $C_{[s,f]} \subseteq [s, f]$. (ii) For all $[a, b] \in \mathcal{F}$, if $[s, f] \cap [a, b] \neq \emptyset$, then we have $C_{[s,f]} \cap [a, b] \neq \emptyset$. By the first property, the subset $C_{[s,f]}$ includes only days in the interval $[s, f]$. By the second property, if the interval $[s, f]$ overlaps with the interval $[a, b]$, then the subset of days $C_{[s,f]}$ preserves this relationship and overlaps with the interval $[a, b]$ as well. A trivial way to construct the intersection preserving subset $C_{[s,f]}$ is to set $C_{[s,f]} = \{s, \dots, f\}$. Using intersection preserving subsets with fewer elements will allow us to obtain better performance guarantees. Later in this section, we elaborate on finding the smallest intersection preserving subsets. Let $\mathcal{C} = \{C_{[s,f]} : [s, f] \in \mathcal{F}\}$ be a collection that includes an intersection preserving subset for each interval $[s, f] \in \mathcal{F}$.

Using this collection, we compute the coefficients $\{\gamma_{i,[s,f]}^q : i \in \mathcal{N}, [s, f] \in \mathcal{F}, q \in \mathcal{Q}\}$ as follows. For all $i \in \mathcal{N}$ and $[s, f] \in \mathcal{F}$, set $\gamma_{i,[s,f]}^{Q+1} = 0$. For each $q = Q, Q-1, \dots, 1$, we execute the two steps.

- **Construct Ideal Assortments:** For all $[s, f] \in \mathcal{F}$, compute the ideal assortment to offer to a customer with a booking request for interval $[s, f]$ as

$$B_{[s,f]}^q = \arg \max_{S \subseteq \mathcal{N}} \left\{ \sum_{i \in \mathcal{N}} \phi_i^q(S) \left[r_{i,[s,f]} - \sum_{[a,b] \in \mathcal{F}} \left| [s, f] \cap C_{[a,b]} \right| \gamma_{i,[a,b]}^{q+1} \right] \right\}. \quad (10)$$

- **Compute Coefficients:** For all $i \in \mathcal{N}$ and $[s, f] \in \mathcal{F}$, compute the coefficient corresponding to resource i and interval $[s, f]$ as

$$\gamma_{i,[s,f]}^q = \lambda_{[s,f]}^q \phi_i^q(B_{[s,f]}^q) \left[r_{i,[s,f]} - \sum_{[a,b] \in \mathcal{F}} \left| [s, f] \cap C_{[a,b]} \right| \gamma_{i,[a,b]}^{q+1} \right] + \gamma_{i,[s,f]}^{q+1}. \quad (11)$$

The algorithm above fully specifies the coefficients $\{\gamma_{i,[s,f]}^q : i \in \mathcal{N}, [s, f] \in \mathcal{F}, q \in \mathcal{Q}\}$, so it also specifies our value function approximations. We can give some intuition for the algorithm.

Given that all resources are available on all days at time period q , $J^q(e)$ corresponds to the optimal total expected revenue obtained from the booking requests arriving over

time periods $\{q, \dots, Q\}$. Our approximation of this optimal total expected revenue is given by $\hat{J}_p^q(e) = \sum_{i \in \mathcal{N}} \sum_{[s,f] \in \mathcal{F}} \gamma_{i,[s,f]}^q$. Thus, we interpret $\gamma_{i,[s,f]}^q$ as an approximation of the optimal total expected revenue over time periods $\{q, \dots, Q\}$ obtained from booking requests for resource i over interval $[s, f]$. In (10), by the definition of an intersection preserving subset, if $[s, f] \cap [a, b] \neq \emptyset$, then $[s, f] \cap C_{[a,b]} \neq \emptyset$ as well. Furthermore, if $[s, f] \cap [a, b] \neq \emptyset$, then accepting a booking request for resource i over interval $[s, f]$ prevents us from accepting a booking request for the same resource over interval $[a, b]$. Therefore, the expression $\sum_{[a,b] \in \mathcal{F}} \left| [s, f] \cap C_{[a,b]} \right| \gamma_{i,[s,f]}^{q+1}$ captures the opportunity cost of booking requests that we can no longer accept by booking resource i for interval $[s, f]$. In this case, $B_{[s,f]}^q$ is the assortment that maximizes the net expected revenue from a customer making a booking request for interval $[s, f]$ at time period q . In (11), we accumulate our approximation to the optimal total expected revenue obtained from the booking requests for resource i over interval $[s, f]$. On the right side of (11), the first term captures the net expected revenue obtained at time period q given that we offer the ideal assortment $B_{[s,f]}^q$, whereas the second term captures the total expected revenue over time periods $\{q+1, \dots, Q\}$.

Although the preceding discussion gives only rough intuition for the algorithm, we can use this algorithm to give a static policy with a performance guarantee. It turns out that this performance guarantee holds for any collection of intersection preserving subsets that we can possibly use in the algorithm. In particular, noting the ideal assortments $\{B_{[s,f]}^q : [s, f] \in \mathcal{F}, q \in \mathcal{Q}\}$ computed through (10), we consider a static policy that always offers the assortment $B_{[s,f]}^q$ of resources to a customer making a booking request for interval $[s, f]$ at time period q . In revenue management, each resource and interval of use combination that can be booked by a customer is referred to as an itinerary. Our polynomial value function approximations have one component for each resource and interval of use combination, so we call this static policy the *itinerary based static policy*. In the next theorem, we give a performance guarantee for the itinerary based static policy. For the collection of intersection preserving subsets $\mathcal{C} = \{C_{[s,f]} : [s, f] \in \mathcal{F}\}$, we define the norm of this collection as $\|\mathcal{C}\| = \max_{[s,f] \in \mathcal{F}} |C_{[s,f]}|$. The performance guarantee that we give for the itinerary based static policy depends on the norm of the collection of intersection preserving subsets that we use in the algorithm. The performance guarantee favors using a collection with a smaller norm.

Theorem 6.1 (Performance of Itinerary Based Static Policy) *The itinerary based static policy obtains at least $1/(1 + \|\mathcal{C}\|)$ fraction of the optimal total expected revenue.*

Similar to the proof of Theorem 4.1, the proof of the theorem above is based on using the value function approximations $\{\hat{J}_p^q : q \in \mathcal{Q}\}$ to give an upper bound on the performance of the optimal policy and a lower bound on the performance of the itinerary based static policy, but

the specifics of proof uses the properties of intersection preserving subsets. We give the proof in Appendix B. Recall that setting $C_{[s,f]} = [s, f]$ trivially yields an intersection preserving subset for the interval $[s, f]$. For this intersection preserving subset, we have $|C_{[s,f]}| = f - s + 1 \leq D_{\max}$, which implies that there exists a collection of intersection preserving subsets whose norm is at most D_{\max} . By Theorem 6.1, the itinerary based static policy that we obtain by using such a trivial collection of intersection preserving subsets has a performance guarantee of $1/(1 + D_{\max})$. In the next theorem, letting $D_{\min} = \min_{[s,f] \in \mathcal{F}} \{f - s + 1\}$ to capture the minimum use duration for any possible booking request, we show that there exists a collection of intersection preserving subsets whose norm does not exceed $1 + \lceil (D_{\max} - 1)/D_{\min} \rceil$. Furthermore, we can find this collection by solving a linear program. In particular, noting that the norm of the collection $\mathcal{C} = \{C_{[s,f]} : [s, f] \in \mathcal{F}\}$ is given by $\|\mathcal{C}\| = \max_{[s,f] \in \mathcal{F}} |C_{[s,f]}|$, using the decision variables $\{C_{[s,f]} : [s, f] \in \mathcal{F}\}$ and t , to find the collection of intersection preserving subsets with the smallest norm, we can solve

$$\begin{aligned} \min \left\{ t : t \geq |C_{[s,f]}| \quad \forall [s, f] \in \mathcal{F}, \right. \\ \left. |C_{[s,f]} \cap [a, b]| \geq 1 \quad \forall [s, f] \text{ and } [a, b] \in \mathcal{F} \text{ such that } [s, f] \cap [a, b] \neq \emptyset, \right. \\ \left. C_{[s,f]} \subseteq [s, f] \quad \forall [s, f] \in \mathcal{F}, \quad t \geq 0 \right\}. \end{aligned} \quad (12)$$

By the first constraint, we have $t = \max_{[s,f] \in \mathcal{F}} |C_{[s,f]}|$ in an optimal solution to problem (12). The second and third constraints ensure that $C_{[s,f]}$ is an intersection preserving subset.

Theorem 6.2 (Smallest Norm Collection) *The optimal objective value of problem (12) is at most $1 + \lceil (D_{\max} - 1)/D_{\min} \rceil$. Furthermore, we can obtain an optimal solution to this problem by solving a linear program with $O(D_{\max}^2 T)$ decision variables and $O(D_{\max}^3 T)$ constraints.*

We give the proof of the above theorem in Appendix C. Thus, by Theorem 6.2, we can solve problem (12) to obtain a collection of intersection preserving subsets whose norm is at most $1 + \lceil (D_{\max} - 1)/D_{\min} \rceil$, in which case, by Theorem 6.1, the corresponding itinerary based static policy has a performance guarantee of $\frac{1}{2 + \lceil (D_{\max} - 1)/D_{\min} \rceil}$. Ma et al. (2020) use nonlinear value function approximations to give a policy with a performance guarantee of $1/(1 + D_{\max})$. Because $D_{\min} \geq 1$, we have $\frac{1}{2 + \lceil (D_{\max} - 1)/D_{\min} \rceil} \geq 1/(1 + D_{\max})$, so the performance guarantee of the itinerary based static policy is at least as good as that of the policy proposed by Ma et al. (2020). As discussed in the introduction, Villa Mahal offers rooms with minimum length of stay requirements, in which case, D_{\min} strictly exceeds one. The performance guarantee for the itinerary based static policy can be significantly better than $1/(1 + D_{\max})$ when D_{\min} is larger than one. Closing this section, we note that the rollout policy from the itinerary based static policy inherits the performance guarantee of $\frac{1}{2 + \lceil (D_{\max} - 1)/D_{\min} \rceil}$ from the itinerary based static policy.

7. Upper Bound on the Optimal Policy Performance

We give an approach to computing an upper bound on the optimal total expected revenue. Such an upper bound becomes useful for assessing the optimality gaps of different policies.

7.1 Resource Based Problems Through Revenue Allocations

One approach to obtaining an upper bound on the optimal total expected revenue is based on solving a linear programming approximation that is formulated under the assumption that the arrivals and choices of the customers take on their expected values; see Gallego et al. (2004), Liu and van Ryzin (2008), and Kunnumkal and Topaloglu (2008). To formulate such an approximation, we use two sets of decision variables. First, we use the decision variable $h_{[s,f]}^q(S)$ to capture the probability of offering the assortment S of resources to a booking request for interval $[s, f]$ at time period q . Second, we use the decision variable $y_{i,[s,f]}^q$ to capture the expected number of bookings for resource i at time period q by a customer interested in making a booking for interval $[s, f]$. In this case, we consider the linear programming approximation

$$\begin{aligned}
Z_{\text{LP}}^* &= \max \sum_{q \in \mathcal{Q}} \sum_{i \in \mathcal{N}} \sum_{[s,f] \in \mathcal{F}} r_{i,[s,f]} y_{i,[s,f]}^q & (13) \\
&\text{st} \sum_{q \in \mathcal{Q}} \sum_{[s,f] \in \mathcal{F}} \sum_{S \subseteq \mathcal{N}} \mathbb{1}_{\{\ell \in [s,f]\}} \phi_i^q(S) h_{[s,f]}^q(S) \leq 1 \quad \forall i \in \mathcal{N}, \ell \in \mathcal{T} \\
&\sum_{S \subseteq \mathcal{N}} h_{[s,f]}^q(S) = \lambda_{[s,f]}^q \quad \forall [s,f] \in \mathcal{F}, q \in \mathcal{Q} \\
&\sum_{S \subseteq \mathcal{N}} \phi_i^q(S) h_{[s,f]}^q(S) = y_{i,[s,f]}^q \quad \forall i \in \mathcal{N}, [s,f] \in \mathcal{F}, q \in \mathcal{Q} \\
&h_{[s,f]}^q(S) \geq 0 \quad \forall [s,f] \in \mathcal{F}, S \subseteq \mathcal{N}, q \in \mathcal{Q},
\end{aligned}$$

where we do not explicitly impose a nonnegativity constraint on the decision variable $y_{i,[s,f]}^q$ because the third and fourth constraints above already ensure this constraint.

In the objective function above, we accumulate the total expected revenue from the bookings. By the first constraint, noting that $\sum_{S \subseteq \mathcal{N}} \phi_i^q(S) h_{[s,f]}^q(S)$ is the expected number of bookings for resource i by a customer arriving at time period q interested in making a booking for interval $[s, f]$, the total expected capacity consumption of each resource on each day does not exceed one. By the second constraint, the total probability that we offer an assortment to a customer arriving at time period q with an interest in making a booking for interval $[s, f]$ is equal to the arrival probability of such a customer. By the third constraint, we compute the expected number of bookings of resource i at time period q by a customer interested in making a booking for interval $[s, f]$. Linear programming approximations are commonly used in the revenue management literature to obtain

upper bounds on the optimal expected revenue. Such upper bounds are known to be asymptotically tight as the capacities of the resources and the expected number of customer arrivals increase linearly at the same rate. This kind of asymptotic tightness result is particularly relevant in airline network revenue management problems, in which the capacities of the flights and the volume of demand served are generally large. In our problem, however, the capacities of all resources are invariably one, even when the number of different resources under consideration is large. In our computational experiments, the upper bound provided by the linear programming approximation can indeed be quite loose. We give a different upper bound that is at least as tight as the one from the linear programming approximation. In our computational experiments, the new upper bound provided by our approach can improve that from the linear programming approximation by as much as 29%. Recall that we refer to each resource and interval of use combination as an itinerary. Thus, for each $i \in \mathcal{N}$ and $[s, f] \in \mathcal{F}$, we have an itinerary $(i, [s, f])$. Our approach is based on allocating the revenue associated with an itinerary to the different resources.

Let $\beta_{i,[s,f] \rightarrow j}^q$ be the portion of the revenue associated with itinerary $(i, [s, f])$ allocated to resource j at time period q . We do not yet specify how we choose the revenue allocations, but if we add the revenue allocations of an itinerary over all resources then we should get the revenue of the itinerary, so the revenue allocations satisfy $\sum_{j \in \mathcal{N}} \beta_{i,[s,f] \rightarrow j}^q = r_{i,[s,f]}$. In our approach, we solve a separate revenue management problem for each resource and collect the value functions for each resource to get an upper bound. In the revenue management problem for resource j , we have limited capacity only for resource j , but infinite capacity for all other resources. Moreover, if we accept a booking request at time period q for itinerary $(i, [s, f])$, then the revenue we collect is the revenue allocation of this itinerary over resource j , which is $\beta_{i,[s,f] \rightarrow j}^q$. We capture the state of resource j by using the vector $\mathbf{x}_j = (x_{j,1}, \dots, x_{j,T}) \in \{0, 1\}^T$, where $x_{j,\ell} = 1$ if and only if resource j is available for use on day ℓ . Noting that $\mathbf{e}_{[s,f]} \in \{0, 1\}^T$ is the vector with ones in the components corresponding to days $\{s, \dots, f\}$, we can find the optimal policy in the revenue management problem for resource j by computing the value functions $\{V_{\beta,j}^q : q \in \mathcal{Q}\}$ through the dynamic program

$$V_{\beta,j}^q(\mathbf{x}_j) = \sum_{[s,f] \in \mathcal{F}} \lambda_{[s,f]}^q \max_{S \subseteq \mathcal{N}} \left\{ \phi_j^q(S) \left(\prod_{\ell=s}^f x_{j,\ell} \right) \left[\beta_{j,[s,f] \rightarrow j}^q + V_{\beta,j}^{q+1}(\mathbf{x}_j - \mathbf{e}_{[s,f]}) - V_{\beta,j}^{q+1}(\mathbf{x}_j) \right] + \sum_{i \in \mathcal{N} \setminus \{j\}} \phi_i^q(S) \beta_{i,[s,f] \rightarrow j}^q \right\} + V_{\beta,j}^{q+1}(\mathbf{x}_j), \quad (14)$$

with the boundary condition that $V_{\beta,j}^{Q+1} = 0$. In the value functions $\{V_{\beta,j}^q : q \in \mathcal{Q}\}$, we make their dependence on $\beta = \{\beta_{i,[s,f] \rightarrow j}^q : i, j \in \mathcal{N}, [s, f] \in \mathcal{F}, q \in \mathcal{Q}\}$ explicit.

To interpret the dynamic program above, at time period q , if we offer the assortment S of resources to a customer making a booking request for interval $[s, f]$, then the customer chooses

resource j with probability $\phi_j^q(S)$. In this case, if we have capacity for resource j to accommodate the booking request, then we make the revenue of $\beta_{j,[s,f] \rightarrow j}^q$. The opportunity cost of the capacities that we consume is $V_{\beta,j}^{q+1}(\mathbf{x}_j) - V_{\beta,j}^{q+1}(\mathbf{x}_j - \mathbf{e}_{[s,f]})$. Because we have infinite capacity for all resources other than resource j , if the customer chooses some other resource i , then we make the revenue of $\beta_{i,[s,f] \rightarrow j}^q$, but the opportunity cost of the capacities that we consume is zero. In the next proposition, we show that we obtain an upper bound on the value functions $\{J^q : q \in \mathcal{Q}\}$ in (1) by solving the dynamic program in (14). We defer the proofs of all results in this section to Appendix D.

Proposition 7.1 (Decomposition Upper Bound) *For any revenue allocations β that satisfy $\sum_{j \in \mathcal{N}} \beta_{i,[s,f] \rightarrow j}^q = r_{i,[s,f]}$ for all $i \in \mathcal{N}$, $[s, f] \in \mathcal{F}$ and $q \in \mathcal{Q}$, letting the value functions $\{V_{\beta,j}^q : q \in \mathcal{Q}\}$ be computed through (14), for each $\mathbf{x} = (\mathbf{x}_j : j \in \mathcal{N}) \in \{0, 1\}^{n \times T}$ and $q \in \mathcal{Q}$, we have*

$$\sum_{j \in \mathcal{N}} V_{\beta,j}^q(\mathbf{x}_j) \geq J^q(\mathbf{x}).$$

Next, we consider the question of choosing the revenue allocations such that our upper bound compares favorably against that from the linear programming approximation.

7.2 Choosing the Revenue Allocations

In problem (13), duplicating the decision variable $h_{[s,f]}^q$ for each resource j to obtain the decision variables $\{h_{[s,f] \rightarrow j}^q : j \in \mathcal{N}\}$, we consider a variant of this problem given by

$$\begin{aligned} \max \quad & \sum_{q \in \mathcal{Q}} \sum_{i \in \mathcal{N}} \sum_{[s,f] \in \mathcal{F}} r_{i,[s,f]} y_{i,[s,f]}^q & (15) \\ \text{st} \quad & \sum_{q \in \mathcal{Q}} \sum_{[s,f] \in \mathcal{F}} \sum_{S \subseteq \mathcal{N}} \mathbf{1}_{\{\ell \in [s,f]\}} \phi_i^q(S) h_{[s,f] \rightarrow i}^q(S) \leq 1 \quad \forall i \in \mathcal{N}, \ell \in \mathcal{T} \\ & \sum_{S \subseteq \mathcal{N}} h_{[s,f] \rightarrow j}^q(S) = \lambda_{[s,f]}^q \quad \forall j \in \mathcal{N}, [s, f] \in \mathcal{F}, q \in \mathcal{Q} \\ & \sum_{S \subseteq \mathcal{N}} \phi_i^q(S) h_{[s,f] \rightarrow j}^q(S) = y_{i,[s,f]}^q \quad \forall i, j \in \mathcal{N}, [s, f] \in \mathcal{F}, q \in \mathcal{Q} \\ & h_{[s,f] \rightarrow j}^q(S) \geq 0 \quad \forall j \in \mathcal{N}, [s, f] \in \mathcal{F}, S \subseteq \mathcal{N}, q \in \mathcal{Q}. \end{aligned}$$

In Lemma D.1 in Appendix D, we show that the problem above has the same optimal objective value as the linear programming approximation in (13). Intuitively speaking, although problem (15) duplicates the decision variable $h_{[s,f]}^q$ in problem (13) for each resource j , it does not duplicate the decision variable $y_{i,[s,f]}^q$. Because the objective functions of both of these two problems depend only on the decision variables $\{y_{i,[s,f]}^q : i \in \mathcal{N}, [s, f] \in \mathcal{F}, q \in \mathcal{Q}\}$, the two problems end up having the same optimal objective value. We use the dual solution to problem (15) to choose the revenue allocations. Associating the dual variables $\{\beta_{i,[s,f] \rightarrow j}^q : i, j \in \mathcal{N}, [s, f] \in \mathcal{F}, q \in \mathcal{Q}\}$ with the third

constraint in problem (15), in the dual of problem (15), the constraint associated with the decision variable $y_{i,[s,f]}^q$ is $\sum_{j \in \mathcal{N}} \beta_{i,[s,f] \rightarrow j}^q = r_{i,[s,f]}$. Thus, letting $\hat{\beta} = \{\hat{\beta}_{i,[s,f] \rightarrow j}^q : i, j \in \mathcal{N}, [s, f] \in \mathcal{F}, q \in \mathcal{Q}\}$ be the optimal values of the dual variables associated with the third constraint in problem (15), we use these optimal values as our revenue allocations. In the next theorem, we show that if we use these revenue allocations in the dynamic program in (14), then we obtain an upper bound on the optimal total expected revenue that is at least as tight as that from the linear program in (13). In this theorem, we use $\mathbf{e}' \in \{0, 1\}^T$ to denote the vector of all ones.

Theorem 7.2 (Choice of Revenue Allocations) *Letting $\hat{\beta}$ be the optimal values of the dual variables associated with the third constraint in problem (15) and the value functions $\{V_{\hat{\beta},j}^q : q \in \mathcal{Q}\}$ be computed through (14) with the revenue allocations $\hat{\beta}$, we have*

$$\sum_{j \in \mathcal{N}} V_{\hat{\beta},j}^1(\mathbf{e}') \leq Z_{\text{LP}}^*.$$

Noting Proposition 7.1, we get $Z_{\text{LP}}^* \geq \sum_{j \in \mathcal{N}} V_{\hat{\beta},j}^1(\mathbf{e}') \geq J^1(\mathbf{e})$, so $\sum_{j \in \mathcal{N}} V_{\hat{\beta},j}^1(\mathbf{e}')$ is an upper bound on the optimal total expected revenue, and this upper bound is at least as tight as that provided by the optimal objective value of the linear programming approximation in (13).

7.3 Computing the Upper Bound

In (14), although we focus only on one resource, the state variable is still high-dimensional, so it is not clear whether we can solve this dynamic program efficiently. We will give an equivalent formulation for this dynamic program by using intervals of days as the state variable, which we will be able to solve efficiently. We use $\mathbf{x}_j = (x_{j,1}, \dots, x_{j,T}) \in \{0, 1\}^T$ to denote the state of resource j , where $x_{j,\ell} = 1$ if and only if resource j is available for use on day ℓ . We refer to the interval $[a, b]$ as a maximal unavailable interval with respect to \mathbf{x}_j if and only if resource j is unavailable on all days in the interval $[a, b]$, but available on days $a - 1$ and $b + 1$; that is, $\mathbf{x}_{j,\ell} = 0$ for all $j \in \{a, \dots, b\}$, $x_{j,a-1} = 1$, and $x_{j,b+1} = 1$. In the revenue management problem for resource j , given that resource j is not available on all days over the maximal unavailable interval $[a, b]$, let $\Gamma_{\beta,j}^q(a, b)$ be the total expected revenue obtained over time periods $\{q, \dots, Q\}$ from the booking requests for intervals that start in the interval $[a, b]$. We can compute the value functions $\{\Gamma_{\beta,j}^q : q \in \mathcal{Q}\}$ as

$$\Gamma_{\beta,j}^q(a, b) = \sum_{k=q}^Q \sum_{[s,f] \in \mathcal{F}} \mathbf{1}_{\{s \in [a,b]\}} \lambda_{[s,f]}^k \max_{S \subseteq \mathcal{N} \setminus \{j\}} \left\{ \sum_{i \in \mathcal{N} \setminus \{j\}} \phi_i^k(S) \beta_{i,[s,f] \rightarrow j}^k \right\}. \quad (16)$$

To interpret (16), in the revenue management problem for resource j , we have infinite capacity for all resources other than resource j . Thus, given that resource j is not available on all days over the interval $[a, b]$, if a customer makes a booking request for an interval $[s, f]$ that starts in the

interval $[a, b]$, then we can offer only assortments not including resource j . If the customer makes a booking for resource i , then the revenue that we obtain is the revenue allocation of itinerary $(i, [s, f])$ to resource j . Because resources other than resource j have infinite capacity and we do not offer resource j , we simply accumulate the expected revenue over time periods $\{q, \dots, Q\}$. On the other hand, recall that we refer to the interval $[a, b]$ as a maximal available interval with respect to \mathbf{x}_j if and only if resource j is available on all days in the interval $[a, b]$, but unavailable on days $a - 1$ and $b + 1$. In the revenue management problem for resource j , given that we do have capacity for resource j over the maximal available interval $[a, b]$, let $\Theta_{\beta, j}^q(a, b)$ be the total expected revenue obtained over time periods $\{q, \dots, Q\}$ from the booking requests for intervals that start in the interval $[a, b]$. We can compute the value functions $\{\Theta_{\beta, j}^q : q \in \mathcal{Q}\}$ as

$$\begin{aligned} \Theta_{\beta, j}^q(a, b) = & \sum_{[s, f] \in \mathcal{F}} \lambda_{[s, f]}^q \mathbf{1}_{\{s \in [a, b], [s, f] \subseteq [a, b]\}} \max_{S \subseteq \mathcal{N}} \left\{ \sum_{i \in \mathcal{N} \setminus \{j\}} \phi_i^q(S) \beta_{i, [s, f] \rightarrow j}^q \right. \\ & \left. + \phi_j^q(S) \left[\beta_{j, [s, f] \rightarrow j}^q + \Theta_{\beta, j}^{q+1}(a, s - 1) + \Theta_{\beta, j}^{q+1}(f + 1, b) + \Gamma_{\beta, j}^{q+1}(s, f) - \Theta_{\beta, j}^{q+1}(a, b) \right] \right\} \\ & + \sum_{[s, f] \in \mathcal{F}} \lambda_{[s, f]}^q \mathbf{1}_{\{s \in [a, b], [s, f] \not\subseteq [a, b]\}} \max_{S \subseteq \mathcal{N} \setminus \{j\}} \left\{ \sum_{i \in \mathcal{N} \setminus \{j\}} \phi_i^q(S) \beta_{i, [s, f] \rightarrow j}^q \right\} + \Theta_{\beta, j}^{q+1}(a, b). \quad (17) \end{aligned}$$

To interpret (17), given that we have capacity for resource j over the interval $[a, b]$, if we have a booking request for an interval $[s, f]$ that starts in $[a, b]$ and the interval $[s, f]$ is included in $[a, b]$, then we have capacity for resource j to serve this booking request. If the customer books resource $i \neq j$, then the revenue that we obtain is the revenue allocation of itinerary $(i, [s, f])$ to resource j , but because the resources other than resource j have infinite capacity, we do not account for the opportunity cost of the capacities that we lose. If, however, the customer books resource j , then we lose the capacity on days $\{s, \dots, f\}$, so the intervals $[a, s - 1]$ and $[f + 1, b]$ become maximal available intervals and the interval $[s, f]$ becomes a maximal unavailable interval. Lastly, if the interval $[s, f]$ is not included in $[a, b]$, then we do not have capacity for resource j to serve a booking request for the interval $[s, f]$, so we can offer only an assortment not including resource j .

Recall that $\mathcal{I}(\mathbf{x}_j)$ is the collection of maximal available intervals with respect to \mathbf{x}_j . Let $\mathcal{H}(\mathbf{x}_j)$ be the collection of maximal unavailable intervals with respect to \mathbf{x}_j . In the top portion of Figure 1, for example, for the value of \mathbf{x}_j in this figure, we have $\mathcal{H}(\mathbf{x}_j) = \{[1, 1], [5, 5], [11, 12]\}$. In the next theorem, we show that we can solve the dynamic program in (17) to get a solution for (14).

Theorem 7.3 (Interval Formulation) *Letting the value function $\{V_{\beta, j}^q : q \in \mathcal{Q}\}$ be computed through (14), for each $\mathbf{x}_j \in \{0, 1\}^T$ and $q \in \mathcal{Q}$, we have*

$$V_{\beta, j}^q(\mathbf{x}_j) = \sum_{[a, b] \in \mathcal{I}(\mathbf{x}_j)} \Theta_{\beta, j}^q(a, b) + \sum_{[a, b] \in \mathcal{H}(\mathbf{x}_j)} \Gamma_{\beta, j}^q(a, b).$$

Next, we give computational experiments to test the quality of our policies and bound.

8. Computational Experiments

We give two sets of computational experiments. The first set is on synthetic datasets that we generate. The second set is on a dataset from an actual boutique hotel.

8.1 Results on Synthetic Datasets

Our experiments are for a boutique hotel. Thus, the set of resources corresponds to a set of unique rooms. Booking resource i over interval $[s, f]$ corresponds to booking room i over days $\{s, \dots, f\}$.

Experimental Setup: We generate our test problems as follows. We have $n = 5$ rooms. The rooms are available for stay during the days indexed by $\mathcal{T} = \{1, \dots, 70\}$, so we have 10 weeks in the booking horizon. Customers arrive over time periods $\mathcal{Q} = \{1, \dots, 700\}$. The set of possible intervals of stay is $\mathcal{F} = \{[a, b] : a \leq b \leq a + D_{\max} - 1\}$, where D_{\max} is the maximum duration of stay parameter we vary. To come up with the revenue $r_{i,[s,f]}$ for stay in room i over interval $[s, f]$, for each $i \in \mathcal{N}$, we sample a base revenue ψ_i from the uniform distribution over $[0, 10]$. Letting $p_{i,\ell}$ be the price of stay in room i on day ℓ , if day ℓ is Friday, Saturday, or Sunday, then we set $p_{i,\ell} = \psi_i$, whereas if day ℓ is Monday, Tuesday, Wednesday, or Thursday, then we set $p_{i,\ell} = \delta \times \psi_i$, where δ is the discount parameter. The revenue for stay in room i over interval $[s, f]$ is $r_{i,[s,f]} = \sum_{\ell=s}^f p_{i,\ell}$.

To come up with the arrival probabilities $\{\lambda_{[s,f]}^q : [s, f] \in \mathcal{F}, q \in \mathcal{Q}\}$, for each $[s, f] \in \mathcal{F}$ and $q \in \mathcal{Q}$, we sample a weight $\beta_{[s,f]}^q$ from the uniform distribution over $[0, \bar{U}_{[s,f]}^q]$ and normalize the weights by setting $\gamma_{[s,f]}^q = \frac{\beta_{[s,f]}^q}{\sum_{[a,b] \in \mathcal{F}} \beta_{[a,b]}^q}$. In this case, using $\{\gamma_{[s,f]}^q : [s, f] \in \mathcal{F}, q \in \mathcal{Q}\}$ as the arrival probabilities, if we always offer all rooms, then the total expected demand for the capacity in all rooms on all days is $\text{Demand} = \sum_{i \in \mathcal{N}} \sum_{q \in \mathcal{Q}} \sum_{[s,f] \in \mathcal{F}} \gamma_{[s,f]}^q \phi_i^q(\mathcal{N}) (f - s + 1)$. Noting that the total capacity available in all rooms on all days is nT , we set the arrival probability for a booking request for interval $[s, f]$ at time period q as $\lambda_{[s,f]}^q = \rho \gamma_{[s,f]}^q \frac{nT}{\text{Demand}}$, where ρ is the load factor parameter we vary. In this case, if we offer all rooms at all time periods, then the ratio between the total expected demand for the capacity and the total available capacity is given by ρ .

Splitting the 700 time periods into three roughly equal segments, we use $\bar{U}_{[s,f]}^q = D_{\max} - (f - s)$ for $q = 1, \dots, 233$, $\bar{U}_{[s,f]}^q = 1$ for $q = 234, \dots, 466$ and $\bar{U}_{[s,f]}^q = f - s + 1$ for $q = 467, \dots, 700$. Thus, the requests for shorter intervals tend to have larger arrival probabilities at the earlier time periods, so we need to carefully reserve the capacity for the requests for longer intervals that tend to arrive later. In this way, we generate test problems that require carefully allocating the available capacity to obtain good performance. Choices of the customers are governed by the multinomial logit model. Thus, using v_i to denote the preference weight of room i and v_0 to denote the preference weight of the no-purchase option, if we offer the assortment S of rooms, then a customer arriving at time period q chooses room i with probability $\phi_i^q(S) = \frac{v_i}{v_0 + \sum_{j \in S} v_j}$. To come up with the preference

weights, for each $i \in \mathcal{N}$, we sample v_i from the uniform distribution over $[0, 1]$ and set $v_0 = \frac{1}{9} \sum_{i \in \mathcal{N}} v_i$, so if we offer all rooms, then a customer leaves without a booking with probability 0.1.

Varying the parameters $D_{\max} \in \{6, 8, 10\}$, $\rho \in \{1.2, 1.6, 2.0\}$, and $\delta \in \{0.7, 0.9\}$, we obtain 18 parameter configurations for our test problems.

Benchmark Policies: Our benchmark policies are based on the linear and polynomial value function approximations, as well as the linear programming approximation.

Linear Approximations (LIN1, LIN5, LINR). We use three benchmark policies based on the linear value function approximations presented in Section 4. By Theorem 4.1, the resource based static policy has a performance guarantee of $1/(2D_{\max})$, but as a static policy, the resource based static policy may offer an unavailable resource for a booking request, which may not be appropriate in practice. To overcome this difficulty, we use the greedy policy with respect to the linear approximations. Replacing $J^{q+1}(\mathbf{x})$ on the right side of (1) with $\hat{J}_L^{q+1}(\mathbf{x}) = \sum_{i \in \mathcal{N}} \sum_{\ell \in \mathcal{T}} \eta_{i,\ell}^{q+1} x_{i,\ell}$, if the state of the system at time period q is \mathbf{x} and there is a booking request for interval $[s, f]$, then the greedy policy with respect to the linear approximations offers the assortment of resources $\hat{S}_L^q(\mathbf{x}) = \arg \max_{S \subseteq \mathcal{N}} \sum_{i \in \mathcal{N}} \phi_i^q(S) \left(\prod_{\ell=s}^f x_{i,\ell} \right) \left(r_{i,[s,f]} - \sum_{\ell=s}^f \eta_{i,\ell}^{q+1} \right)$. By the discussion given immediately after (1), this policy never offers an unavailable resource. Using the same outline in the proof of Theorem 4.1, we can show that the greedy policy with respect to the linear approximations also has a performance guarantee of $1/(2D_{\max})$. The key is to observe that an analogue of the chain of inequalities in the proof of Proposition 5.5 still holds when we replace $A_{[s,f]}^q$ with $\hat{S}_L^q(\mathbf{x})$.

We use LIN1 to refer to the greedy policy with respect to the linear approximations, where we emphasize the fact that this policy computes the opportunity costs $\{\eta_{i,\ell}^q : i \in \mathcal{N}, \ell \in \mathcal{T}, q \in \mathcal{Q}\}$ once at the beginning of the selling horizon. We also use a version of LIN1 that splits the selling horizon into five equal segments to recompute the opportunity costs at the beginning of each segment based on the state of the system at the beginning of the segment. In particular, if the state of the system at the beginning of the segment is \mathbf{x} , then we set $\eta_{i,\ell}^q = 0$ for all $i \in \mathcal{N}$, $\ell \in \mathcal{T}$, and $q \in \mathcal{Q}$ such that $x_{i,\ell} = 0$. We compute the other opportunity costs using the algorithm in Section 4. We refer to this policy as LIN5. Lastly, we use the rollout policy from the resource based static policy. We refer to this policy as LINR. By the discussion at the end of Section 3, this policy inherits the performance guarantee of $1/(2D_{\max})$ from the resource based static policy. Furthermore, this policy never offers an unavailable resource for a booking request. Observe that LIN1 and LIN5 are greedy policies with respect to linear approximations $\{J_L^q : q \in \mathcal{Q}\}$, whereas noting that $\{J_\mu^q : q \in \mathcal{Q}\}$ in (5) is not separable, LINR does not necessarily use a separable approximation.

Polynomial Approximations (POL1, POL5, POLR). The benchmark policies POL1, POL5, and POLR are the analogues of LIN1, LIN5, and LINR, but they use the polynomial value function

approximations in Section 6. As in LIN1, POL1 is the greedy policy with respect to the polynomial approximations. We can show that it shares the guarantee of $\frac{1}{2 + \lceil (D_{\max} - 1) / D_{\min} \rceil}$ with the itinerary based static policy. In POL5, when recomputing the coefficients $\{\gamma_{i,[s,f]}^q : i \in \mathcal{N}, [s,f] \in \mathcal{F}, q \in \mathcal{Q}\}$ at the beginning of a segment, if the state of the system is \mathbf{x} , then we set $\gamma_{i,[s,f]}^q = 0$ for all $i \in \mathcal{N}$, $[s,f] \in \mathcal{F}$, and $q \in \mathcal{Q}$ such that $\prod_{\ell=s}^f x_{i,\ell} = 0$. We compute the other coefficients using the algorithm in Section 6. Lastly, POLR is the rollout of the itinerary based static policy.

Linear Programming Approximation (LP1, LP5, LPR). We use three benchmark policies based on the linear program in (13). The first constraint in problem (13) ensures that the total expected capacity consumption of each resource on each day does not exceed the capacity of one. Letting $\{\theta_{i,\ell} : i \in \mathcal{N}, \ell \in \mathcal{T}\}$ be the optimal values of the dual variables associated with the first constraint, we use $\theta_{i,\ell}$ as the opportunity cost of the capacity of resource i on day ℓ , yielding the value function approximations $\{\hat{J}_{\text{LP}}^q : q \in \mathcal{Q}\}$ with $\hat{J}_{\text{LP}}^q(\mathbf{x}) = \sum_{i \in \mathcal{N}} \sum_{\ell \in \mathcal{T}} \theta_{i,\ell} x_{i,\ell}$. Similar to LIN1, if the state of the system at time period q is \mathbf{x} and we have a booking request for interval $[s,f]$, then we offer the assortment of resources $\hat{S}_{\text{LP}}^q(\mathbf{x}) = \arg \max_{S \subseteq \mathcal{N}} \sum_{i \in \mathcal{N}} \phi_i^q(S) (\prod_{\ell=s}^f x_{i,\ell}) (r_{i,[s,f]} - \sum_{\ell=s}^f \theta_{i,\ell})$. We refer to this policy as LP1. Thus, LP1 corresponds to the greedy policy with respect to the linear approximations $\{\hat{J}_{\text{LP}}^q : q \in \mathcal{Q}\}$, but the opportunity costs in these linear approximations are obtained from the dual solution to the linear programming approximation.

We also use a version of LP1 that splits the selling horizon into five equal segments to recompute the opportunity costs at the beginning of each segment. If the segment starts at time period \bar{q} with the state \mathbf{x} , then we solve problem (13) after replacing the set of time periods \mathcal{Q} with $\{\bar{q}, \dots, Q\}$ and the right side of the first constraint with $\{x_{i,\ell} : i \in \mathcal{N}, \ell \in \mathcal{T}\}$. Letting $\{\theta_{i,\ell} : i \in \mathcal{N}, \ell \in \mathcal{T}\}$ be the optimal values of the dual variables associated with the first constraint, we use $\theta_{i,\ell}$ as the opportunity cost of the capacity of resource i on day ℓ . We use LP5 to refer to this policy. Lastly, we extract a static policy from problem (13) and use the rollout policy from this static policy. In our static policy, letting $\{\hat{h}_{[s,f]}^q(S) : [s,f] \in \mathcal{F}, S \subseteq \mathcal{N}, q \in \mathcal{Q}\}$ and $\{\hat{y}_{i,[s,f]}^q : i \in \mathcal{N}, [s,f] \in \mathcal{F}, q \in \mathcal{Q}\}$ be an optimal solution to problem (13), if we have a booking request for interval $[s,f]$ at time period q , then we offer the assortment S of resources with probability $\hat{h}_{[s,f]}^q(S) / \lambda_{[s,f]}^q$. We use the rollout policy from this static policy and refer to the resulting rollout policy as LPR.

Dynamic Programming Decomposition (DEC). This benchmark policy is the standard dynamic programming decomposition method in the revenue management literature; see, for example, Liu and van Ryzin (2004). In this policy, we heuristically decompose the dynamic programming formulation in (1) by each resource and day combination. Note that each resource has one unit of capacity on each day. Therefore, this policy constructs linear value function approximations $\{\hat{J}_{\text{Dec}}^q : q \in \mathcal{Q}\}$ of the form $\hat{J}_{\text{Dec}}^q(\mathbf{x}) = \sum_{i \in \mathcal{N}} \sum_{\ell \in \mathcal{T}} \zeta_{i,\ell}^q x_{i,\ell}$ for some opportunity costs

Params. (D_{\max}, ρ, δ)	% Gap in Bounds	Params. (D_{\max}, ρ, δ)	% Gap in Bounds	Params. (D_{\max}, ρ, δ)	% Gap in Bounds
(6, 1.2, 0.9)	15.38	(8, 1.2, 0.9)	25.62	(10, 1.2, 0.9)	13.82
(6, 1.2, 0.7)	11.99	(8, 1.2, 0.7)	15.53	(10, 1.2, 0.7)	12.93
(6, 1.6, 0.9)	20.10	(8, 1.6, 0.9)	29.73	(10, 1.6, 0.9)	26.50
(6, 1.6, 0.7)	24.53	(8, 1.6, 0.7)	23.35	(10, 1.6, 0.7)	28.08
(6, 2.0, 0.9)	24.90	(8, 2.0, 0.9)	25.84	(10, 2.0, 0.9)	26.75
(6, 2.0, 0.7)	22.86	(8, 2.0, 0.7)	16.09	(10, 2.0, 0.7)	26.33
Avg.	19.96	Avg.	22.69	Avg.	22.40

Table 1 Percent gap between the upper bounds for the synthetic datasets.

$\{\zeta_{i,\ell}^q : i \in \mathcal{N}, \ell \in \mathcal{T}, q \in \mathcal{Q}\}$, in which case, we can use the greedy policy with respect to these linear value function approximations. We refer to this policy as DEC.

Lastly, the policy in Baek and Ma (2019) was not competitive to our linear or polynomial approximations. We give detailed comparisons with Baek and Ma (2019) in Appendix E.

Comparison of Upper Bounds: The optimal objective value Z_{LP}^* of the linear programming approximation in (13) provides an upper bound on the optimal total expected revenue. By Proposition 7.1, we can solve the dynamic program in (14) to also obtain an upper bound on the optimal total expected revenue. Furthermore, by Theorem 7.2, we can use the optimal dual solution to problem (15) to choose the revenue allocations such that the upper bound we obtain is at least as tight as that provided by Z_{LP}^* . In particular, letting $\hat{\beta}$ be the optimal values of the dual variables associated with the third constraint in problem (15), $\sum_{j \in \mathcal{N}} V_{\hat{\beta},j}^1(e')$ is an upper bound on the optimal total expected revenue, and this upper bound is at least as tight as Z_{LP}^* . In Table 1, we compare the upper bounds Z_{LP}^* and $\sum_{j \in \mathcal{N}} V_{\hat{\beta},j}^1(e')$. In the table, the first column shows the parameter configuration for each test problem. The second column shows the percent gap $100 \times \frac{Z_{LP}^* - \sum_{j \in \mathcal{N}} V_{\hat{\beta},j}^1(e')}{\sum_{j \in \mathcal{N}} V_{\hat{\beta},j}^1(e')}$ between the two upper bounds.

Our results indicate that the upper bounds from our approach can dramatically improve those from the linear programming approximation. The gaps can reach 29.73%. In our approach, once we allocate the revenue associated with each itinerary over different resources, the revenue management problem that we solve for each resource incorporates the uncertainty in the customer arrivals and choices, giving our approach an edge over the linear programming approximation. It is known that DEC also provides an upper bound on the optimal total expected revenue. In our test problems, the upper bounds from DEC were virtually identical to those from the linear programming approximation. The upper bound from DEC is based on decomposing the problem by each resource and day combination, whereas the upper bound from our approach is based on decomposing the problem by each resource. Therefore, intuitively speaking, our approach captures the interactions between the different days more faithfully than DEC.

Policy Performance: Considering the performance of the benchmark policies given earlier in this section, we estimate the total expected revenue obtained by each policy by simulating

Param. (D_{\max}, ρ, δ)	Total Exp. Revenue									
	LIN1	POL1	LP1	LIN5	POL5	LP5	LINR	POLR	LPR	DEC
(6, 1.2, 0.9)	79.23	75.19	76.99	80.89	75.34	78.03	81.05	80.29	81.11	77.27
(6, 1.2, 0.7)	70.17	68.57	68.53	71.27	69.81	69.67	73.00	73.04	73.97	67.65
(6, 1.6, 0.9)	74.94	67.84	72.44	77.10	68.53	74.56	78.60	75.96	77.92	70.58
(6, 1.6, 0.7)	80.52	72.77	76.25	82.78	75.25	82.24	85.02	83.07	84.52	80.59
(6, 2.0, 0.9)	79.14	70.24	75.93	81.44	72.63	80.30	85.04	82.84	84.14	78.76
(6, 2.0, 0.7)	77.16	67.61	69.50	79.28	69.34	74.21	80.85	77.39	79.28	73.91
(8, 1.2, 0.9)	75.27	66.18	73.88	78.14	69.27	75.91	78.23	77.84	79.29	75.79
(8, 1.2, 0.7)	70.39	65.71	67.03	72.16	66.50	68.56	71.07	71.09	73.44	68.41
(8, 1.6, 0.9)	80.24	67.56	77.76	83.06	69.40	82.51	84.64	81.96	84.82	79.77
(8, 1.6, 0.7)	76.84	67.39	73.59	80.14	68.72	77.55	79.75	78.52	81.57	77.08
(8, 2.0, 0.9)	80.54	66.96	73.47	81.41	68.68	78.11	84.26	81.37	81.56	79.38
(8, 2.0, 0.7)	73.90	60.85	71.99	77.28	62.89	74.74	79.21	73.62	77.22	75.12
(10, 1.2, 0.9)	68.61	62.56	66.24	70.72	63.42	67.95	73.89	71.43	74.33	70.04
(10, 1.2, 0.7)	69.95	66.56	67.02	69.90	68.48	68.98	73.09	72.90	75.02	68.76
(10, 1.6, 0.9)	73.74	63.60	72.25	76.37	67.13	74.55	79.11	74.98	78.40	72.09
(10, 1.6, 0.7)	77.31	62.45	73.58	79.96	65.11	78.69	82.22	78.10	81.86	78.61
(10, 2.0, 0.9)	78.43	63.80	75.57	81.09	65.45	78.78	83.65	79.14	81.83	78.43
(10, 2.0, 0.7)	78.72	62.28	73.18	81.39	63.77	80.55	83.93	75.64	82.01	80.84
Avg.	75.84	66.56	72.51	78.02	68.32	75.88	79.81	77.18	79.57	75.17

Table 2 Total expected revenues obtained by the benchmark policies for the synthetic datasets.

the decisions of the policy over 100 sample paths. In Table 2, we compare the total expected revenues obtained by the benchmark policies. In the table, the first column shows the parameter configuration for each test problem. In the second to fourth columns, we focus on the performance of LIN1, POL1, and LP1, which compute the coefficients of the value function approximations once at the beginning of the selling horizon. In the fifth to seventh columns, we focus on the performance of LIN5, POL5, and LP5, which compute the coefficients of the value function approximations five times over the selling horizon. In the eighth to tenth columns, we focus on the performance of LINR, POLR, and LPR, which use a rollout policy from a static policy. In the eleventh column, we focus on the performance of DEC. For each benchmark policy, we express its total expected revenue as a percentage of the upper bound on the optimal total expected revenue. In other words, using Rev to denote the total expected revenue of a benchmark policy and UB to denote the upper bound, we report $100 \times \frac{\text{Rev}}{\text{UB}}$. We use the upper bound obtained from the dynamic program in (14), along with the revenue allocations provided by an optimal dual solution to problem (15).

Regarding the benchmark policies LIN1, POL1, and LP1, which compute the coefficients of the value function approximations once at the beginning of the selling horizon, our results indicate that LIN1 consistently provides better performance than the other two policies. When we compute the coefficients of the value function approximations five times over the selling horizon, on average, the total expected revenues obtained by LIN5, POL5, and LP5 improve those obtained by LIN1, POL1, and LP1, respectively, by 2.86%, 2.66%, and 4.61%. The performance of LIN5 is still noticeably better than that of POL5 and LP5. When we use the rollout policies, on average, the total expected revenues obtained by LINR, POLR, and LPR improve those obtained by LIN5, POL5, and LP5, respectively, by

2.30%, 13.08%, and 4.96%. Noting the performance of improvement of POLR over POL5, even if the static policy that we use is not a superior policy, performing rollout on the static policy can dramatically improve the performance of the static policy. Overall, irrespective of whether the static policy is obtained by constructing value function approximations or by using an optimal solution to a linear programming approximation, our rollout approach can be quite effective in obtaining even better policies. The performance of DEC lags behind all rollout policies, which is encouraging, because dynamic programming decomposition is known to be one of the strongest benchmarks for revenue management problems. So, rollout policies, such as LINR, can perform noticeably better than the strongest benchmark, while having a performance guarantee. In Appendix E, we compare our policies with the policy in Baek and Ma (2019). On average, LINR, POLR, and LPR improve on the policy in Baek and Ma (2019), respectively, by 20.83%, 18.19%, and 20.63%.

Another useful feature of rollout policies is that we can perform ensemble rollout on a collection of static policies. In particular, letting $\{\mu_k : k = 1, \dots, K\}$ be a collection of K static policies, we define the value function approximations $\{\hat{J}_{\max}^q : q \in \mathcal{Q}\}$ as $\hat{J}_{\max}^q(\mathbf{x}) = \max_{k=1, \dots, K} J_{\mu_k}^q(\mathbf{x})$, where $\{J_{\mu_k}^q : q \in \mathcal{Q}\}$ are the value functions of the static policy μ^k computed through (2). In this case, the total expected revenue from the greedy policy with respect to the value function approximations $\{\hat{J}_{\max}^q : q \in \mathcal{Q}\}$ is at least as large as the total expected revenue from *each* of the static policies $\{\mu_k : k = 1, \dots, K\}$; see Example 2.3.2 in Bertsekas (2012). Thus, we can obtain a policy that is at least as good as each of the static policies. We performed such an ensemble rollout on the collection of resource based static policy, itinerary based static policy, and static policy from problem (13). In our test problems, the performance of the ensemble rollout policy was statistically indistinguishable from the best rollout policy in Table 2 for each test problem.

We carry out all of our computational experiments in Java 1.8.0 with 16 GB of RAM and 2.8 GHz Intel Core i7 CPU. For our largest test problems with $D_{\max} = 10$, the average CPU time to compute the linear and polynomial value function approximations are, respectively, 16.12 and 30.04 seconds. The average CPU time to compute our upper bound is 4.02 hours. The average CPU time to perform rollout on a static policy is 5.12 minutes.

8.2 Results on a Boutique Hotel Dataset

We test the performance of our benchmark policies and upper bounds by using the reservation data from an actual boutique hotel.

Experimental Setup: We have access to the booking data from a boutique hotel for reservations made between May 13, 2020 and September 8, 2020. The bookings are for stays from

June 1, 2020 to October 31, 2020. The hotel has six unique rooms. The price of each room and day pair is pre-fixed, but the hotel changes the availability of rooms based on real-time information on booked capacities. In total, the dataset contains 157 bookings. With the exception of three bookings, all bookings are for one to eight days. We dropped those three bookings in our estimation procedure. We estimate the parameters of our model as follows.

There are 119 days between May 13 and September 8, so the customers arrive over a selling horizon of 119 days. We divide each day into 10 time periods. In this case, the set of time periods is $\mathcal{Q} = \{1, \dots, 1190\}$. For each $q \in \mathcal{Q}$, we use $\text{day}(q)$ to denote the calendar date corresponding to time period q . On the other hand, there are 153 days between June 1 and October 31, so we use $\mathcal{T} = \{1, \dots, 153\}$ to index the days of stay. For each $\ell \in \mathcal{T}$, we use $\overline{\text{day}}(\ell)$ to denote the calendar date corresponding to day ℓ . To estimate the arrival probabilities for the booking requests, we proceed with the following two assumptions. First, at each time period, there is a fixed probability that we have a request for a booking for a certain number of days into the future. Second, given that we have a booking request, there is a fixed probability that the booking is for a certain length of stay. In particular, we let θ_k be the probability that we have a booking request for k days into the future. Given that we have a booking request, we let η_d be the probability that this booking request is for d days. In this case, the probability that we have a booking request for interval $[s, f]$ at time period q is given by $\lambda_{[s,f]}^q = \theta_{\overline{\text{day}}(s) - \text{day}(q)} \times \eta_{f-s+1}$. Because there are 171 days between May 13 and October 31, a customer can make a booking as many as 171 days in advance. Thus, we need to estimate the parameter θ_k for all $k = 1, \dots, 171$. Noting that the dataset contains only 157 bookings, we divide the interval $\{1, \dots, 171\}$ into the seven subintervals $[1, 3]$, $[4, 7]$, $[8, 14]$, $[15, 28]$, $[29, 42]$, $[43, 56]$, and $[57, 171]$, assuming that the value of θ_k is constant when k takes values in each of these intervals. When estimating the parameters $\{\theta_k : k = 1, \dots, 171\}$, we impose the constraint that $\sum_{k=1}^{171} \theta_k \leq 1$, in which case we have $\sum_{[s,f] \in \mathcal{F}} \lambda_{[s,f]}^q \leq 1$ for all $q \in \mathcal{Q}$. We use the multinomial logit model to capture the choice process among rooms. Thus, letting v_i be the preference weight of room i and normalizing the preference weight of the no-purchase option to one, the choice probability of room i within the assortment S is $\phi_i^q(S) = \frac{v_i}{1 + \sum_{j \in S} v_j}$.

We use maximum likelihood estimation to estimate the parameters of our model. In Appendix F, we give the details of our estimation procedure.

Computational Results: To broaden our experimental setup, we scaled the arrival probabilities $\{\lambda_{[s,f]}^q : [s, f] \in \mathcal{F}, q \in \mathcal{Q}\}$ by a constant to obtain test problems with load factors taking values in $\{0.8, 1.2, 1.6, 2.0\}$. We measure the load factor in the same way we do for our test problems on the synthetic datasets. We give our results in Table 3. In this table, the first column shows the load factor. The second column shows the percent gap between the upper bounds obtained by using

Load Fact.	Gap in Bounds	Total Exp. Revenue									
		LIN1	POL1	LP1	LIN5	POL5	LP5	LINR	POLR	LPR	DEC
0.8	12.25	81.50	83.69	81.41	82.59	83.33	82.05	83.96	84.43	83.00	82.10
1.2	10.10	85.76	87.73	85.69	87.34	87.75	86.69	88.18	88.70	87.16	86.43
1.6	8.15	87.19	89.35	87.47	88.16	89.44	87.58	90.18	89.69	88.85	87.40
2.0	6.65	88.90	90.82	88.79	89.96	90.85	88.78	91.03	91.22	90.53	88.80
Avg.	9.29	85.84	87.90	85.84	87.01	87.84	86.28	88.34	88.51	87.38	86.18

Table 3 Computational results for the boutique hotel datasets.

the dynamic program in (14) and the linear programming approximation in (13). The remaining columns give the total expected revenues obtained by our benchmark policies, each expressed as a percentage of the upper bound on the optimal total expected revenue. Our results indicate that the upper bounds provided by our approach significantly improve those from the linear programming approximation. The gaps in the upper bounds can exceed 12%. When compared with our results on the synthetic datasets, LIN5, POL5, and LP5 obtain more modest improvements in the total expected revenues by recomputing the coefficients of the value function approximations five times over the selling horizon, instead of once. On the other hand, similar to our results on the synthetic datasets, the rollout policies, especially LINR and POLR, perform noticeably better than DEC. There are test problems in which the total expected revenue improvements over DEC provided by our rollout policies can reach 3.08%. Thus, our rollout approach continues to provide effective policies for the datasets based on actual boutique hotel bookings.

9. Conclusion

We developed policies with performance guarantees by exploiting two features of the underlying problem. First, each resource is unique, so we have one unit of capacity for each resource. Second, customers book resources for consecutive days. We hope our study inspires other research on leveraging features of the underlying problem. In our model, a customer makes a booking request for an interval, in response to which we offer an assortment of resources. This model is consistent with the booking systems of many boutique hotels and freelancer matching platforms, but it would be interesting to develop a model that also allows customers to specify a window of possible use. In this case, the decision would be assortments of resources to offer over different possible intervals of use. Our analysis of the itinerary based static policy goes through, but our efficient rollout result does not readily extend and more work is needed. Lastly, the performance guarantee of $1/(1 + \|\mathcal{C}\|)$ for the itinerary based static policy reveals a relationship between policy performance and pattern of resource usage. It would be interesting to explore such relationships in other problems.

References

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

- Aouad, A., V. Farias, R. Levi. 2020. Assortment optimization under consider-then-choose choice models. *Management Science* (forthcoming).
- Baek, J., W. Ma. 2019. Bifurcating constraints to improve approximation ratios for network revenue management with reusable resources. Tech. rep., Columbia University, New York, NY.
- Bertsekas, D. P. 2012. *Dynamic Programming and Optimal Control*, vol. 2. 4th ed. Athena Scientific, Belmont, MA.
- Bertsekas, D. P. 2017. *Dynamic Programming and Optimal Control*, vol. 1. 4th ed. Athena Scientific, Belmont, MA.
- Bertsekas, D. P., D. A. Castanon. 1999. Rollout algorithms for stochastic scheduling problems. *Journal of Heuristics* **5**(1) 89–108.
- Bertsekas, D. P., J. N. Tsitsiklis, C. Wu. 1997. Rollout algorithms for combinatorial optimization. *Journal of Heuristics* **3**(3) 245–262.
- Bertsimas, D., I. Popescu. 2003. Revenue management in a dynamic network environment. *Transportation Science* **37**(3) 257–277.
- Blanchet, J., G. Gallego, V. Goyal. 2016. A Markov chain approximation to choice modeling. *Operations Research* **64**(4) 886–905.
- Boutique Homes. 2020. Boutique homes: Very small hotels. URL <https://www.boutique-homes.com/small-hotels/>. Last access date: December 8, 2020.
- Bront, J. J. M., I. Mendez Diaz, G. Vulcano. 2009. A column generation algorithm for choice-based network revenue management. *Operations Research* **57**(3) 769–784.
- Cao, Y., P. Rusmevichientong, H. Topaloglu. 2020. Revenue management under a mixture of multinomial logit and independent demand models. Tech. rep., Cornell University, New York, NY.
- Chaneton, J. M., G. Vulcano. 2011. Computing bid prices for revenue management under customer choice behavior. *Manufacturing & Service Operations Management* **13**(4) 452–470.
- Cooper, W. L. 2002. Asymptotic behavior of an allocation policy for revenue management. *Operations Research* **50**(4) 720–727.
- Davis, J. M., G. Gallego, H. Topaloglu. 2014. Assortment optimization under variants of the nested logit model. *Operations Research* **62**(2) 250–273.
- Gallego, G., G. Iyengar, R. Phillips, A. Dubey. 2004. Managing flexible products on a network. Available at SSRN: <https://ssrn.com/abstract=3567371>.
- Jasin, S., S. Kumar. 2012. A re-solving heuristic with bounded revenue loss for network revenue management with customer choice. *Mathematics of Operations Research* **37**(2) 313–345.
- Jasin, S., S. Kumar. 2013. Analysis of deterministic LP-based booking limit and bid price controls for revenue management. *Operations Research* **61**(6) 1312–1320.
- Kirshner, S. N., M. Nediak. 2015. Scalable dynamic bid prices for network revenue management in continuous time. *Production and Operations Management* **24**(10) 1621–1635.
- Kunnumkal, S., K. Talluri. 2016a. On a piecewise-linear approximation for network revenue management. *Mathematics of Operations Research* **41**(1) 72–91.
- Kunnumkal, S., K. Talluri. 2016b. Technical note: A note on relaxations of choice network revenue management dynamic program. *Operations Research* **64**(1) 158–166.

- Kunnumkal, S., H. Topaloglu. 2008. A refined deterministic linear program for the network revenue management problem with customer choice behavior. *Naval Research Logistics* **55**(6) 563–580.
- Kunnumkal, S., H. Topaloglu. 2010. A new dynamic programming decomposition method for the network revenue management problem with customer choice behavior. *Production and Operations Management* **19**(5) 575–590.
- Liu, Q., G. J. van Ryzin. 2008. On the choice-based linear programming model for network revenue management. *Manufacturing & Service Operations Management* **10**(2) 288–310.
- 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. 2020. Efficient policies for efficient volunteer crowdsourcing. *Management Science* (forthcoming).
- Meissner, J., A. Strauss, K. Talluri. 2012. An enhanced concave program relaxation for choice network revenue management. *Production and Operations Management* **22**(1) 71–87.
- Nemhauser, G. L., L. A. Wolsey. 1988. *Integer and Combinatorial Optimization*. John Wiley & Sons, New York, NY.
- Rusmevichientong, P., M. Sumida, H. Topaloglu. 2020. Dynamic assortment optimization for reusable products with random usage durations. *Management Science* **66**(7) 2820–2844.
- Secomandi, N. 2001. A rollout policy for the vehicle routing problem with stochastic demands. *Operations Research* **49**(5) 796–802.
- Talluri, K., G. J. van Ryzin. 1998. An analysis of bid-price controls for network revenue management. *Management Science* **44**(11-part-1) 1577–1593.
- Talluri, K., G. J. van Ryzin. 2004. Revenue management under a general discrete choice model of consumer behavior. *Management Science* **50**(1) 15–33.
- Tong, C., H. Topaloglu. 2013. On the approximate linear programming approach for network revenue management problems. *INFORMS Journal on Computing* **26**(1) 121–134.
- Topaloglu, H. 2008. A stochastic approximation method to compute bid prices in network revenue management problems. *INFORMS Journal on Computing* **20**(4) 596–610.
- Topaloglu, H. 2009. Using Lagrangian relaxation to compute capacity-dependent bid prices in network revenue management. *Operations Research* **57**(3) 637–649.
- van Ryzin, G. J., G. Vulcano. 2008. Computing virtual nesting controls for network revenue management under customer choice behavior. *Manufacturing & Service Operations Management* **10**(3) 448–467.
- Villa Mahal. 2020. Villa Mahal Hotel website. URL <https://www.villamahahal.com>. Last access date: December 8, 2020.
- Vossen, T. W. M., D. Zhang. 2015a. A dynamic disaggregation approach to approximate linear programs for network revenue management. *Production and Operations Management* **24**(3) 469–487.
- Vossen, T. W. M., D. Zhang. 2015b. Reductions of approximate linear programs for network revenue management. *Operations Research* **63**(6) 1352–1371.
- Zhang, D., D. Adelman. 2009. An approximate dynamic programming approach to network revenue management with customer choice. *Transportation Science* **43**(3) 381–394.
- Zhang, D., W. L. Cooper. 2005. Revenue management for parallel flights with customer choice behavior. *Operations Research* **53**(3) 415–431.
- Zhang, D., W. L. Cooper. 2009. Pricing substitutable flights in airline revenue management. *European Journal of Operational Research* **197**(3) 848–861.

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

Online Appendix

Revenue Management with Heterogeneous Resources: Unit Resource Capacities, Advance Bookings, and Itineraries over Time Intervals

Appendix A: Offering Itineraries Only with Positive Contributions

We establish a lemma that we use throughout the paper to argue that some of our policies never offer an unavailable resource. In the next lemma, for fixed $\{p_i : i \in \mathcal{N}\}$, we focus on the problem

$$\max_{S \subseteq \mathcal{N}} \left\{ \sum_{i \in \mathcal{N}} \phi_i^q(S) p_i \right\}. \quad (18)$$

Lemma A.1 (Offering Only Positive Contributions) *There exists an optimal solution S^* to problem (18) such that $S^* \subseteq \{i \in \mathcal{N} : p_i > 0\}$.*

Proof: Letting S^* be an optimal solution to problem (18), we define the assortment $\hat{S} \subseteq S^*$ as $\hat{S} = \{i \in S^* : p_i > 0\}$. By the assumption that $\phi_i^q(S \cup \{j\}) \leq \phi_i^q(S)$ for all $i \in S$ and $j \notin S$, we have $\phi_i^q(S^*) \leq \phi_i^q(\hat{S})$ for all $i \in \hat{S}$. In this case, using the fact that $p_i \leq 0$ for all $i \in S^* \setminus \hat{S}$, along with $p_i > 0$ for all $i \in \hat{S}$, we obtain the chain of inequalities

$$\sum_{i \in S^*} \phi_i^q(S^*) p_i = \sum_{i \in \hat{S}} \phi_i^q(S^*) p_i + \sum_{i \in S^* \setminus \hat{S}} \phi_i^q(S^*) p_i \leq \sum_{i \in \hat{S}} \phi_i^q(S^*) p_i \leq \sum_{i \in \hat{S}} \phi_i^q(\hat{S}) p_i.$$

Thus, the chain of inequalities above shows that \hat{S} is also an optimal solution to problem (18). Furthermore, we have $\hat{S} \subseteq \{i \in \mathcal{N} : p_i > 0\}$. ■

Appendix B: Performance Guarantee for Itinerary Based Static Policy

In this section, we give a proof for Theorem 6.1. We need two preliminary lemmas. The next lemma is the analogue of Lemma 5.2. Its proof is identical to that of Lemma 5.2 and omitted.

Lemma B.1 (Nonnegative Contribution of Each Resource) *For each $i \in \mathcal{N}$, $[s, f] \in \mathcal{F}$ and $q \in \mathcal{Q}$, we have $\phi_i^q(B_{[s,f]}^q) [r_{i,[s,f]} - \sum_{[a,b] \in \mathcal{F}} | [s, f] \cap C_{[a,b]} | \gamma_{i,[a,b]}^{q+1}] \geq 0$.*

In the next lemma, we give an upper bound on the opportunity cost of the capacities consumed by using resource i to serve a request for interval $[s, f]$ under the polynomial approximations.

Lemma B.2 (Upper Bound on Opportunity Cost) *For each $i \in \mathcal{N}$, $[s, f] \in \mathcal{F}$ and $\mathbf{x} \in \{0, 1\}^{n \times T}$ such that $\prod_{\ell=s}^f x_{i,\ell} = 1$, we have $\hat{J}_p^q(\mathbf{x}) - \hat{J}_p^q(\mathbf{x} - \mathbf{e}_{i,[s,f]}) \leq \sum_{[a,b] \in \mathcal{F}} | [s, f] \cap C_{[a,b]} | \gamma_{i,[a,b]}^q$.*

Proof: Using the previous lemma, by (11), we have $\gamma_{i,[s,f]}^q \geq 0$ for all $i \in \mathcal{N}$ and $[s, f] \in \mathcal{F}$. Using the fact that $\hat{J}_p^q(\mathbf{x}) = \sum_{i \in \mathcal{N}} \sum_{[s,f] \in \mathcal{F}} \gamma_{i,[s,f]}^q \prod_{\ell=s}^f x_{i,\ell}$, the difference $\hat{J}_p^q(\mathbf{x}) - \hat{J}_p^q(\mathbf{x} - \mathbf{e}_{i,[s,f]})$ is

$$\begin{aligned} \hat{J}_p^q(\mathbf{x}) - \hat{J}_p^q(\mathbf{x} - \mathbf{e}_{i,[s,f]}) &= \sum_{[a,b] \in \mathcal{F}} \gamma_{i,[a,b]}^q \mathbf{1}_{\{[a,b] \cap [s,f] \neq \emptyset\}} \left(\prod_{\ell=a}^b x_{i,\ell} \right) \\ &\leq \sum_{[a,b] \in \mathcal{F}} \gamma_{i,[a,b]}^q \mathbf{1}_{\{[a,b] \cap [s,f] \neq \emptyset\}} \leq \sum_{[a,b] \in \mathcal{F}} |[s, f] \cap C_{[a,b]}| \gamma_{i,[a,b]}^q, \end{aligned}$$

where the last inequality holds because $C_{[a,b]}$ is intersection preserving, so if $[a, b] \cap [s, f] \neq \emptyset$, then $[s, f] \cap C_{[a,b]} \neq \emptyset$. In this case, we have $\mathbf{1}_{\{[a,b] \cap [s,f] \neq \emptyset\}} \leq |[s, f] \cap C_{[a,b]}|$. ■

In the next proposition, we show that we can use the value function approximations $\{\hat{J}_p^q : q \in \mathcal{Q}\}$ to come up with an upper bound on the optimal total expected revenue.

Proposition B.3 (Upper Bound on Optimal Performance) *Noting that the optimal total expected revenue is $J^1(\mathbf{e})$, we have $J^1(\mathbf{e}) \leq (1 + \|\mathcal{C}\|) \hat{J}_p^1(\mathbf{e})$.*

Proof: Letting $\beta_{i,\ell}^q = \sum_{[a,b] \in \mathcal{F}} \mathbf{1}_{\{\ell \in C_{[a,b]}\}} \gamma_{i,[a,b]}^q$ for notational brevity, we define the linear value function approximations $\{\hat{V}_p^q : q \in \mathcal{Q}\}$ as $\hat{V}_p^q(\mathbf{x}) = \hat{J}_p^q(\mathbf{e}) + \sum_{i \in \mathcal{N}} \sum_{\ell \in \mathcal{T}} \beta_{i,\ell}^q x_{i,\ell}$. We have

$$\begin{aligned} &\sum_{[s,f] \in \mathcal{F}} \lambda_{[s,f]}^q \max_{S \subseteq \mathcal{N}} \left\{ \sum_{i \in \mathcal{N}} \phi_i^q(S) \left(\prod_{\ell=s}^f x_{i,\ell} \right) \left[r_{i,[s,f]} + \hat{V}_p^{q+1}(\mathbf{x} - \mathbf{e}_{i,[s,f]}) - \hat{V}_p^{q+1}(\mathbf{x}) \right] \right\} \\ &= \sum_{[s,f] \in \mathcal{F}} \lambda_{[s,f]}^q \max_{S \subseteq \mathcal{N}} \left\{ \sum_{i \in \mathcal{N}} \phi_i^q(S) \left(\prod_{\ell=s}^f x_{i,\ell} \right) \left[r_{i,[s,f]} - \sum_{h=s}^f \beta_{i,h}^{q+1} \right] \right\} \\ &\stackrel{(a)}{\leq} \sum_{[s,f] \in \mathcal{F}} \lambda_{[s,f]}^q \max_{S \subseteq \mathcal{N}} \left\{ \sum_{i \in \mathcal{N}} \phi_i^q(S) \left[r_{i,[s,f]} - \sum_{h=s}^f \beta_{i,h}^{q+1} \right] \right\} \\ &\stackrel{(b)}{=} \sum_{[s,f] \in \mathcal{F}} \lambda_{[s,f]}^q \max_{S \subseteq \mathcal{N}} \left\{ \sum_{i \in \mathcal{N}} \phi_i^q(S) \left[r_{i,[s,f]} - \sum_{[a,b] \in \mathcal{F}} |C_{[a,b]} \cap [s, f]| \gamma_{i,[a,b]}^{q+1} \right] \right\} \\ &\stackrel{(c)}{=} \sum_{[s,f] \in \mathcal{F}} \lambda_{[s,f]}^q \sum_{i \in \mathcal{N}} \phi_i^q(B_{[s,f]}^q) \left[r_{i,[s,f]} - \sum_{[a,b] \in \mathcal{F}} |C_{[a,b]} \cap [s, f]| \gamma_{i,[a,b]}^{q+1} \right] \\ &\stackrel{(d)}{=} \sum_{i \in \mathcal{N}} \sum_{[s,f] \in \mathcal{F}} (\gamma_{i,[s,f]}^q - \gamma_{i,[s,f]}^{q+1}) = \hat{J}_p^q(\mathbf{e}) - \hat{J}_p^{q+1}(\mathbf{e}) \\ &\stackrel{(e)}{\leq} \hat{J}_p^q(\mathbf{e}) - \hat{J}_p^{q+1}(\mathbf{e}) + \sum_{i \in \mathcal{N}} \sum_{\ell \in \mathcal{T}} (\beta_{i,\ell}^q - \beta_{i,\ell}^{q+1}) x_{i,\ell} = \hat{V}_p^q(\mathbf{x}) - \hat{V}_p^{q+1}(\mathbf{x}). \end{aligned}$$

Here, (a) follows from the same argument that we use to obtain the inequality (a) in the proof of Proposition 5.4. In (b), we use the definition of $\beta_{i,\ell}^q$ and use the interchange of sums $\sum_{h=s}^f \sum_{[a,b] \in \mathcal{F}} \mathbf{1}_{\{h \in C_{[a,b]}\}} = \sum_{[a,b] \in \mathcal{F}} \sum_{h=s}^f \mathbf{1}_{\{h \in C_{[a,b]}\}} = \sum_{[a,b] \in \mathcal{F}} |C_{[a,b]} \cap [s, f]|$. Also, (c) follows from (10), and (d) follows from (11). To see that (e) holds, note that Lemma B.1, along with (11), implies that $\gamma_{i,[s,f]}^q \geq \gamma_{i,[s,f]}^{q+1}$, in which case, we also have $\beta_{i,\ell}^q \geq \beta_{i,\ell}^{q+1}$. The chain of inequalities above shows

that the value function approximations $\{\hat{V}_p^q : q \in \mathcal{Q}\}$ satisfy (8), in which case, by the discussion that follows (8), we have $\hat{V}_p^q(\mathbf{x}) \geq J^q(\mathbf{x})$ for all $\mathbf{x} \in \{0, 1\}^{n \times T}$ and $q \in \mathcal{Q}$. Thus, we have

$$\begin{aligned} J^1(\mathbf{e}) &\leq \hat{V}_p^1(\mathbf{e}) = \hat{J}_p^1(\mathbf{e}) + \sum_{i \in \mathcal{N}} \sum_{\ell \in \mathcal{T}} \beta_{i,\ell}^1 = \hat{J}_p^1(\mathbf{e}) + \sum_{i \in \mathcal{N}} \sum_{[a,b] \in \mathcal{F}} \sum_{\ell \in \mathcal{T}} \mathbf{1}_{\{\ell \in C_{[a,b]}\}} \gamma_{i,[a,b]}^1 \\ &= \hat{J}_p^1(\mathbf{e}) + \sum_{i \in \mathcal{N}} \sum_{[a,b] \in \mathcal{F}} |C_{[a,b]}| \gamma_{i,[a,b]}^1 \leq \hat{J}_p^1(\mathbf{e}) + \|\mathcal{C}\| \sum_{i \in \mathcal{N}} \sum_{[a,b] \in \mathcal{F}} \gamma_{i,[a,b]}^1 = (1 + \|\mathcal{C}\|) \hat{J}_p^1(\mathbf{e}), \end{aligned}$$

where the last inequality holds because $\|\mathcal{C}\| = \max_{[a,b] \in \mathcal{F}} |C_{[a,b]}|$, and the last equality follows from the fact that $\hat{J}_p^q(\mathbf{e}) = \sum_{i \in \mathcal{N}} \sum_{[a,b] \in \mathcal{F}} \gamma_{i,[a,b]}^q$. \blacksquare

Let $U_p^q(\mathbf{x})$ be the total expected revenue obtained by the itinerary based static policy over time periods $\{q, \dots, Q\}$ given that the state of the system at time period q is \mathbf{x} . We can compute the value functions $\{U_p^q : q \in \mathcal{Q}\}$ through the dynamic program in (9) after replacing the ideal assortments $\{A_{[s,f]}^q : [s,f] \in \mathcal{F}, q \in \mathcal{Q}\}$ with $\{B_{[s,f]}^q : [s,f] \in \mathcal{F}, q \in \mathcal{Q}\}$. In the next proposition, we lower bound the performance of the itinerary based static policy.

Proposition B.4 (Lower Bound on Policy Performance) *Letting $\{U_p^q : q \in \mathcal{Q}\}$ be the value functions of the itinerary based static policy, for each $\mathbf{x} \in \{0, 1\}^{n \times T}$ and $q \in \mathcal{Q}$, $U_p^q(\mathbf{x}) \geq \hat{J}_p^q(\mathbf{x})$.*

Proof: We show the result by using induction over the time periods. At time period $Q+1$, since we have $U_p^{Q+1} = 0 = \hat{J}_p^{Q+1}$, the inequality holds at time period $Q+1$. Assuming that the inequality holds at time period $q+1$, we show that the inequality holds at time period q as well. By the induction hypothesis, $U_p^{q+1}(\mathbf{x})$ and $U_p^{q+1}(\mathbf{x} - \mathbf{e}_{i,[s,f]})$ are, respectively, lower bounded by $\hat{J}_p^{q+1}(\mathbf{x})$ and $\hat{J}_p^{q+1}(\mathbf{x} - \mathbf{e}_{i,[s,f]})$, so by (9), we get

$$\begin{aligned} U_p^q(\mathbf{x}) &\geq \sum_{[s,f] \in \mathcal{F}} \lambda_{[s,f]}^q \left\{ \sum_{i \in \mathcal{N}} \phi_i^q(B_{[s,f]}^q) \left(\prod_{\ell=s}^f x_{i,\ell} \right) \left[r_{i,[s,f]} + \hat{J}_p^{q+1}(\mathbf{x} - \mathbf{e}_{i,[s,f]}) - \hat{J}_p^{q+1}(\mathbf{x}) \right] \right\} + \hat{J}_p^{q+1}(\mathbf{x}) \\ &\stackrel{(a)}{\geq} \sum_{[s,f] \in \mathcal{F}} \lambda_{[s,f]}^q \left\{ \sum_{i \in \mathcal{N}} \phi_i^q(B_{[s,f]}^q) \left(\prod_{\ell=s}^f x_{i,\ell} \right) \left[r_{i,[s,f]} - \sum_{[a,b] \in \mathcal{F}} | [s,f] \cap C_{[a,b]} | \gamma_{i,[a,b]}^{q+1} \right] \right\} + \hat{J}_p^{q+1}(\mathbf{x}) \\ &\stackrel{(b)}{=} \sum_{i \in \mathcal{N}} \sum_{[s,f] \in \mathcal{F}} (\gamma_{i,[s,f]}^q - \gamma_{i,[s,f]}^{q+1}) \left(\prod_{\ell=s}^f x_{i,\ell} \right) + \hat{J}_p^{q+1}(\mathbf{x}) \stackrel{(c)}{=} [\hat{J}_p^q(\mathbf{x}) - \hat{J}_p^{q+1}(\mathbf{x})] + \hat{J}_p^{q+1}(\mathbf{x}) = \hat{J}_p^q(\mathbf{x}), \end{aligned}$$

where (a) follows from Lemma B.2, (b) holds by (11), and (c) holds by the definition of $\hat{J}_p^q(\mathbf{x})$. The chain of inequalities above establishes the result. \blacksquare

We can use Propositions B.3 and B.4 to give a proof for Theorem 6.1.

Proof of Theorem 6.1:

By Proposition B.3, we have $\hat{J}_p^1(\mathbf{e}) \geq \frac{1}{1+\|\mathcal{C}\|} J^1(\mathbf{e})$. Using Proposition B.4 with $\mathbf{x} = \mathbf{e}$ and $q = 1$, we have $U_p^1(\mathbf{e}) \geq \hat{J}_p^1(\mathbf{e})$. So, we get $U_p^1(\mathbf{e}) \geq \hat{J}_p^1(\mathbf{e}) \geq \frac{1}{1+\|\mathcal{C}\|} J^1(\mathbf{e})$. \blacksquare

Appendix C: Norm of a Collection of Intersection Preserving Subsets

In this section, we give a proof for Theorem 6.2. To see that the first statement holds, we construct a feasible solution to problem (12) that provides an objective value of $1 + \lceil (D_{\max} - 1)/D_{\min} \rceil$. In particular, we set $\hat{t} = 1 + \lceil (D_{\max} - 1)/D_{\min} \rceil$. Furthermore, for each $[s, f] \in \mathcal{F}$, we set

$$\hat{C}_{[s,f]} = \left\{ s + k D_{\min} : k = 0, 1, 2, \dots, \left\lceil \frac{f-s}{D_{\min}} \right\rceil - 1 \right\} \cup \{f\}. \quad (19)$$

The solution $\{\hat{C}_{[s,f]} : [s, f] \in \mathcal{F}\}$ and \hat{t} provides an objective value of $\hat{t} = 1 + \lceil (D_{\max} - 1)/D_{\min} \rceil$ for problem (12). We proceed to arguing that this solution is feasible for problem (12). Using the fact that $f - s + 1 \leq D_{\max}$ for all $[s, f] \in \mathcal{F}$, we have $|\hat{C}_{[s,f]}| \leq \left\lceil \frac{f-s}{D_{\min}} \right\rceil + 1 \leq \left\lceil \frac{D_{\max}-1}{D_{\min}} \right\rceil + 1 = \hat{t}$. Thus, the first constraint is satisfied. The smallest element of $\hat{C}_{[s,f]}$ is s . Noting that $s + \left(\left\lceil \frac{f-s}{D_{\min}} \right\rceil - 1 \right) D_{\min} \leq s + \frac{f-s}{D_{\min}} D_{\min} = f$, none of the elements of $\hat{C}_{[s,f]}$ exceeds f , so $\hat{C}_{[s,f]} \subseteq [s, f]$. Thus, the third constraint is satisfied. To check the second constraint, consider any $[a, b] \in \mathcal{F}$ such that $[s, f] \cap [a, b] \neq \emptyset$. We show that $\hat{C}_{[s,f]} \cap [a, b] \neq \emptyset$. We consider three cases.

First, consider the case $[s, f] \subseteq [a, b]$. Because $\hat{C}_{[s,f]} \subseteq [s, f]$ by the earlier discussion in the proof, we get $\hat{C}_{[s,f]} \subseteq [s, f] \subseteq [a, b]$, so $\hat{C}_{[s,f]} \cap [a, b] \neq \emptyset$, as desired.

Second, consider the case $[s, f] \not\subseteq [a, b]$ and $[s, f] \not\supseteq [a, b]$. Because $[s, f] \cap [a, b] \neq \emptyset$, we must have $f \in [a, b]$ or $s \in [a, b]$. Noting that $s \in \hat{C}_{[s,f]}$ and $f \in \hat{C}_{[s,f]}$, we get $\hat{C}_{[s,f]} \cap [a, b] \neq \emptyset$, as desired.

Third, consider the case $[s, f] \supseteq [a, b]$. To get a contradiction, suppose, on the contrary, that $\hat{C}_{[s,f]} \cap [a, b] = \emptyset$. Since $[s, f] \supseteq [a, b]$, we have $s \leq a \leq b \leq f$, but noting that $\hat{C}_{[s,f]} \cap [a, b] = \emptyset$ and $s, f \in \hat{C}_{[s,f]}$, there are two successive days c_1, c_2 in $\hat{C}_{[s,f]}$ such that $c_1 < a \leq b < c_2$. Thus, there are at least $b - a + 1$ days in between days c_1 and c_2 . Because $b - a + 1 \geq D_{\min}$, there must be at least D_{\min} days in between days c_1 and c_2 . On the other hand, by our construction of $\hat{C}_{[s,f]}$ in (19), there are at most $D_{\min} - 1$ days in between two successive days in $\hat{C}_{[s,f]}$, which is a contradiction! Therefore, the first statement in the theorem holds.

To see that the second statement holds, we write the second constraint in problem (12) as $|C_{[s,f]} \cap [a, b]| \geq \mathbb{1}_{\{[s,f] \cap [a,b] \neq \emptyset\}}$ for all $[s, f], [a, b] \in \mathcal{F}$. Define the constant $Z_{[s,f]}^*$ as

$$Z_{[s,f]}^* = \min \left\{ |C_{[s,f]}| : |C_{[s,f]} \cap [a, b]| \geq \mathbb{1}_{\{[s,f] \cap [a,b] \neq \emptyset\}} \quad \forall [a, b] \in \mathcal{F}, C_{[s,f]} \subseteq [s, f] \right\}, \quad (20)$$

where the only decision variable is $C_{[s,f]}$. In this case, problem (12) becomes equivalent to $\min\{t : t \geq Z_{[s,f]}^* \quad \forall [s, f] \in \mathcal{F}\}$, which has the optimal objective value $\max_{[s,f] \in \mathcal{F}} Z_{[s,f]}^*$.

In the rest of the proof, we will show that we can compute $Z_{[s,f]}^*$ in (20) by solving a minimization linear program with $O(D_{\max})$ decision variables and $O(D_{\max}^2)$ constraints. Thus, letting χ^* be the

optimal objective value of problem (12), by the discussion in the previous paragraph, we have $\chi^* = \max_{[s,f] \in \mathcal{F}} Z_{[s,f]}^*$, so χ^* is the maximum of the optimal objective values of $|\mathcal{F}| = O(D_{\max} T)$ minimization linear programs, each having $O(D_{\max})$ decision variables and $O(D_{\max}^2)$ constraints. In this case, it immediately follows that we can compute the maximum of the optimal objective values of these linear programs by solving a single linear program with $O(|\mathcal{F}| D_{\max}) = O(D_{\max}^2 T)$ decision variables and $O(|\mathcal{F}| D_{\max}^2) = O(D_{\max}^3 T)$ constraints. To compute $Z_{[s,f]}^*$ in (20) by solving a linear program, we use the decision variables $\{x_\ell : \ell = s, \dots, f\} \in \{0, 1\}^{f-s+1}$, where $x_\ell = 1$ if and only if day ℓ is included in the intersection preserving subset $C_{[s,f]}$. We write problem (20) as

$$\begin{aligned} \min \quad & \sum_{\ell=s}^f x_\ell \\ \text{st} \quad & \sum_{\ell=s}^f \mathbf{1}_{\{\ell \in [a,b]\}} x_\ell \geq \mathbf{1}_{\{[s,f] \cap [a,b] \neq \emptyset\}} \quad \forall [a,b] \in \mathcal{F} \\ & x_\ell \in \{0, 1\} \quad \forall \ell = s, \dots, f. \end{aligned}$$

Each row of the constraint matrix above includes only consecutive ones. Such a matrix is called an interval matrix and it is totally unimodular; see Corollary 2.10 in Chapter III.1 in Nemhauser and Wolsey (1988). Thus, we can relax the integrality requirements without an integrality gap. Also, the problem above has a covering constraint and the right side of the constraint never exceeds one, which implies that even if we did not have an upper bound of one on the decision variables, these decision variables would never take a value greater than one. Thus, we can drop the constraints $x_\ell \leq 1$ for all $\ell = s, \dots, f$. Lastly, the right side of the constraint is nonzero for all $[a,b] \in \mathcal{F}$ such that $[s,f] \cap [a,b] \neq \emptyset$ and there are only $O(D_{\max}^2)$ such constraints. Thus, the problem above actually has $f - s + 1 = O(D_{\max})$ decision variables and $O(D_{\max}^2)$ constraints.

Appendix D: Upper Bound on the Optimal Policy Performance

In this section, we give the proofs of the three results in Section 7, along with Lemma D.1 that we use in that section. Here is the proof of Proposition 7.1.

Proof of Proposition 7.1:

We will use a simple manipulation in the proof. In particular, for fixed values $\{\varphi_i : i \in \mathcal{N}\}$, we have the chain of equalities

$$\sum_{j \in \mathcal{N}} \varphi_j \sum_{i \in \mathcal{N} \setminus \{j\}} \beta_{j,[s,f] \rightarrow i}^q = \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N} \setminus \{i\}} \varphi_j \beta_{j,[s,f] \rightarrow i}^q = \sum_{j \in \mathcal{N}} \sum_{i \in \mathcal{N} \setminus \{j\}} \varphi_i \beta_{i,[s,f] \rightarrow j}^q, \quad (21)$$

where the first equality holds by interchanging the order of sums and the second equality holds by interchanging the roles of i and j . We show the proposition by using induction over the time

periods. At time period $Q + 1$, we have $J^{Q+1} = 0 = \sum_{j \in \mathcal{N}} V_{\beta,j}^{Q+1}$, so the result holds at time period $Q + 1$. Assuming that the result holds at time period $q + 1$, we show that the result holds at time period q as well. Fixing $\mathbf{x} \in \{0, 1\}^{n \times T}$ and $q \in \mathcal{Q}$, let $S_{[s,f]}^*$ be an optimal solution to the maximization problem on the right side of (1). By the induction hypothesis, $J^{q+1}(\mathbf{x} - \mathbf{e}_{j,[s,f]}) \leq \sum_{i \in \mathcal{N} \setminus \{j\}} V_{\beta,i}^{q+1}(\mathbf{x}_i) + V_{\beta,j}^{q+1}(\mathbf{x}_j - \mathbf{e}_{[s,f]})$ and $J^{q+1}(\mathbf{x}) \leq \sum_{j \in \mathcal{N}} V_{\beta,j}^{q+1}(\mathbf{x}_j)$. Using (1), we get

$$\begin{aligned}
J^q(\mathbf{x}) &= \sum_{[s,f] \in \mathcal{F}} \lambda_{[s,f]}^q \sum_{j \in \mathcal{N}} \phi_j^q(S_{[s,f]}^*) \left(\prod_{\ell=s}^f x_{j,\ell} \right) \left[r_{j,[s,f]} + J^{q+1}(\mathbf{x} - \mathbf{e}_{j,[s,f]}) \right] + J^{q+1}(\mathbf{x}) \\
&\stackrel{(a)}{\leq} \sum_{[s,f] \in \mathcal{F}} \lambda_{[s,f]}^q \sum_{j \in \mathcal{N}} \phi_j^q(S_{[s,f]}^*) \left(\prod_{\ell=s}^f x_{j,\ell} \right) \left[r_{j,[s,f]} + V_{\beta,j}^{q+1}(\mathbf{x}_j - \mathbf{e}_{[s,f]}) \right] + \sum_{j \in \mathcal{N}} V_{\beta,j}^{q+1}(\mathbf{x}_j) \\
&\stackrel{(b)}{=} \sum_{[s,f] \in \mathcal{F}} \lambda_{[s,f]}^q \left\{ \sum_{j \in \mathcal{N}} \phi_j^q(S_{[s,f]}^*) \left(\prod_{\ell=s}^f x_{j,\ell} \right) \left[\beta_{j,[s,f] \rightarrow j}^q + V_{\beta,j}^{q+1}(\mathbf{x}_j - \mathbf{e}_{[s,f]}) - V_{\beta,j}^{q+1}(\mathbf{x}_j) \right] \right. \\
&\quad \left. + \sum_{j \in \mathcal{N}} \sum_{i \in \mathcal{N} \setminus \{j\}} \phi_i^q(S_{[s,f]}^*) \left(\prod_{\ell=s}^f x_{i,\ell} \right) \beta_{i,[s,f] \rightarrow j}^q \right\} + \sum_{j \in \mathcal{N}} V_{\beta,j}^{q+1}(\mathbf{x}_j) \\
&\stackrel{(c)}{=} \sum_{j \in \mathcal{N}} \left\{ \sum_{[s,f] \in \mathcal{F}} \lambda_{[s,f]}^q \left\{ \phi_j^q(S_{[s,f]}^*) \left(\prod_{\ell=s}^f x_{j,\ell} \right) \left[\beta_{j,[s,f] \rightarrow j}^q + V_{\beta,j}^{q+1}(\mathbf{x}_j - \mathbf{e}_{[s,f]}) - V_{\beta,j}^{q+1}(\mathbf{x}_j) \right] \right. \right. \\
&\quad \left. \left. + \sum_{i \in \mathcal{N} \setminus \{j\}} \phi_i^q(S_{[s,f]}^*) \left(\prod_{\ell=s}^f x_{i,\ell} \right) \beta_{i,[s,f] \rightarrow j}^q \right\} + V_{\beta,j}^{q+1}(\mathbf{x}_j) \right\} \\
&\leq \sum_{j \in \mathcal{N}} \left\{ \sum_{[s,f] \in \mathcal{F}} \lambda_{[s,f]}^q \max_{S \subseteq \mathcal{N}} \left\{ \phi_j^q(S) \left(\prod_{\ell=s}^f x_{j,\ell} \right) \left[\beta_{j,[s,f] \rightarrow j}^q + V_{\beta,j}^{q+1}(\mathbf{x}_j - \mathbf{e}_{[s,f]}) - V_{\beta,j}^{q+1}(\mathbf{x}_j) \right] \right. \right. \\
&\quad \left. \left. + \sum_{i \in \mathcal{N} \setminus \{j\}} \phi_i^q(S) \left(\prod_{\ell=s}^f x_{i,\ell} \right) \beta_{i,[s,f] \rightarrow j}^q \right\} + V_{\beta,j}^{q+1}(\mathbf{x}_j) \right\} \\
&\stackrel{(d)}{\leq} \sum_{j \in \mathcal{N}} \left\{ \sum_{[s,f] \in \mathcal{F}} \lambda_{[s,f]}^q \max_{S \subseteq \mathcal{N}} \left\{ \phi_j^q(S) \left(\prod_{\ell=s}^f x_{j,\ell} \right) \left[\beta_{j,[s,f] \rightarrow j}^q + V_{\beta,j}^{q+1}(\mathbf{x}_j - \mathbf{e}_{[s,f]}) - V_{\beta,j}^{q+1}(\mathbf{x}_j) \right] \right. \right. \\
&\quad \left. \left. + \sum_{i \in \mathcal{N} \setminus \{j\}} \phi_i^q(S) \beta_{i,[s,f] \rightarrow j}^q \right\} + V_{\beta,j}^{q+1}(\mathbf{x}_j) \right\} \\
&\stackrel{(e)}{=} \sum_{j \in \mathcal{N}} V_{\beta,j}^q(\mathbf{x}_j).
\end{aligned}$$

In the chain of inequalities above, (a) uses the induction hypothesis. To see that (b) holds, we note that $r_{j,[s,f]} = \sum_{i \in \mathcal{N}} \beta_{j,[s,f] \rightarrow i}^q = \beta_{j,[s,f] \rightarrow j}^q + \sum_{i \in \mathcal{N} \setminus \{j\}} \beta_{j,[s,f] \rightarrow i}^q$ and use (21) after identifying φ_j with $\phi_j^q(S_{[s,f]}^*) \prod_{\ell=s}^f x_{j,\ell}$. Also, (c) holds by rearranging the order of the sums. To get (d), we use the same argument that we use to obtain inequality (a) in the proof of Proposition 5.4. Lastly, (e) follows from (14). The chain of inequalities above completes the induction argument. \blacksquare

Next, we state and prove Lemma D.1.

Lemma D.1 (Equivalence of Linear Programs) *Problems (13) and (15) have the same optimal objective value.*

Proof: We let $\widehat{\mathbf{h}} = \left\{ \widehat{h}_{[s,f]}^q(S) : S \subseteq \mathcal{N}, [s,f] \in \mathcal{F}, q \in \mathcal{Q} \right\}$ and $\widehat{\mathbf{y}} = \left\{ \widehat{y}_{i,[s,f]} : i \in \mathcal{N}, [s,f] \in \mathcal{F} \right\}$ be an optimal solution to problem (13). For each $j \in \mathcal{N}$, we set $\widetilde{h}_{[s,f] \rightarrow j}^q(S) = \widehat{h}_{[s,f]}^q(S)$. In this case, we observe that the solution $\widetilde{\mathbf{h}} = \left\{ \widetilde{h}_{[s,f] \rightarrow j}^q(S) : j \in \mathcal{N}, S \subseteq \mathcal{N}, [s,f] \in \mathcal{F}, q \in \mathcal{Q} \right\}$ and $\widehat{\mathbf{y}}$ is feasible to problem (15) and provides the same objective value as the optimal objective value of problem (13). Therefore, the optimal objective value of problem (15) is at least as large as that of problem (13). In the rest of the proof, we show that the reverse inequality also holds. We let $\widetilde{\mathbf{h}} = \left\{ \widetilde{h}_{[s,f] \rightarrow j}^q(S) : j \in \mathcal{N}, S \subseteq \mathcal{N}, [s,f] \in \mathcal{F}, q \in \mathcal{Q} \right\}$ and $\widetilde{\mathbf{y}} = \left\{ \widetilde{y}_{i,[s,f]} : i \in \mathcal{N}, [s,f] \in \mathcal{F} \right\}$ be an optimal solution to problem (15). We define $\widehat{h}_{[s,f]}^q(S)$ as

$$\widehat{h}_{[s,f]}^q(S) = \frac{1}{n} \sum_{j \in \mathcal{N}} \widetilde{h}_{[s,f] \rightarrow j}^q(S).$$

We establish that the solution $\widehat{\mathbf{h}} = \left\{ \widehat{h}_{[s,f]}^q(S) : S \subseteq \mathcal{N}, [s,f] \in \mathcal{F}, q \in \mathcal{Q} \right\}$ and $\widetilde{\mathbf{y}}$ is feasible to problem (13). Using the definition of $\widehat{h}_{[s,f]}^q(S)$ above, we have

$$\sum_{S \subseteq \mathcal{N}} \widehat{h}_{[s,f]}^q(S) = \frac{1}{n} \sum_{j \in \mathcal{N}} \sum_{S \subseteq \mathcal{N}} \widetilde{h}_{[s,f] \rightarrow j}^q(S) \stackrel{(a)}{=} \frac{1}{n} \sum_{j \in \mathcal{N}} \lambda_{[s,f]}^q = \lambda_{[s,f]}^q,$$

where (a) holds because the solution $(\widetilde{\mathbf{h}}, \widetilde{\mathbf{y}})$ satisfies the second constraint in problem (15). Thus, the solution $(\widehat{\mathbf{h}}, \widetilde{\mathbf{y}})$ satisfies the second constraint in problem (13).

To check that the solution $(\widehat{\mathbf{h}}, \widetilde{\mathbf{y}})$ satisfies the third constraint in problem (13), using the definition of $\widehat{h}_{[s,f]}^q(S)$ once more, we have

$$\sum_{S \subseteq \mathcal{N}} \phi_i^q(S) \widehat{h}_{[s,f]}^q(S) = \frac{1}{n} \sum_{j \in \mathcal{N}} \sum_{S \subseteq \mathcal{N}} \phi_i^q(S) \widetilde{h}_{[s,f] \rightarrow j}^q(S) \stackrel{(b)}{=} \frac{1}{n} \sum_{j \in \mathcal{N}} \widetilde{y}_{i,[s,f]}^q = \widetilde{y}_{i,[s,f]}^q, \quad (22)$$

where (b) holds because the solution $(\widetilde{\mathbf{h}}, \widetilde{\mathbf{y}})$ satisfies the third constraint in problem (15). Thus, the solution $(\widehat{\mathbf{h}}, \widetilde{\mathbf{y}})$ satisfies the third constraint in problem (13).

Lastly, we check that the solution $(\widehat{\mathbf{h}}, \widetilde{\mathbf{y}})$ satisfies the first constraint in problem (13). In particular, we have the chain of inequalities

$$\begin{aligned} \sum_{q \in \mathcal{Q}} \sum_{[s,f] \in \mathcal{F}} \sum_{S \subseteq \mathcal{N}} \mathbf{1}_{\{\ell \in [s,f]\}} \phi_i^q(S) \widehat{h}_{[s,f]}^q(S) &= \sum_{q \in \mathcal{Q}} \sum_{[s,f] \in \mathcal{F}} \mathbf{1}_{\{\ell \in [s,f]\}} \sum_{S \subseteq \mathcal{N}} \phi_i^q(S) \widehat{h}_{[s,f]}^q(S) \\ &\stackrel{(c)}{=} \sum_{q \in \mathcal{Q}} \sum_{[s,f] \in \mathcal{F}} \mathbf{1}_{\{\ell \in [s,f]\}} \widetilde{y}_{i,[s,f]}^q \stackrel{(d)}{=} \sum_{q \in \mathcal{Q}} \sum_{[s,f] \in \mathcal{F}} \mathbf{1}_{\{\ell \in [s,f]\}} \sum_{S \subseteq \mathcal{N}} \phi_i^q(S) \widetilde{h}_{[s,f] \rightarrow i}^q(S) \stackrel{(e)}{\leq} 1, \end{aligned}$$

where (c) follows from (22), (d) follows from the fact that the solution $(\widetilde{\mathbf{h}}, \widetilde{\mathbf{y}})$ satisfies the third constraint in problem (15) with $i = j$, and (e) holds because the solution $(\widetilde{\mathbf{h}}, \widetilde{\mathbf{y}})$ also satisfies the first

constraint in problem (15). Therefore, the solution $(\hat{\mathbf{h}}, \tilde{\mathbf{y}})$ satisfies the first constraint in problem (13). Thus, the solution $(\hat{\mathbf{h}}, \tilde{\mathbf{y}})$ is feasible to problem (13). The solution $(\hat{\mathbf{h}}, \tilde{\mathbf{y}})$ provides the objective value $\sum_{q \in \mathcal{Q}} \sum_{i \in \mathcal{N}} \sum_{[s,f] \in \mathcal{F}} r_{i,[s,f]} \tilde{y}_{i,[s,f]}^q$, which is the optimal objective value of problem (15). So, the optimal objective value of problem (13) is at least as large as that of problem (15). ■

To write the dual of problem (15), associating the dual variables $\boldsymbol{\mu} = \{\mu_{i,\ell} : i \in \mathcal{N}, \ell \in \mathcal{T}\}$, $\boldsymbol{\sigma} = \{\sigma_{[s,f] \rightarrow j}^q : j \in \mathcal{N}, [s,f] \in \mathcal{F}, q \in \mathcal{Q}\}$ and $\boldsymbol{\beta} = \{\beta_{i,[s,f] \rightarrow j}^q : i, j \in \mathcal{N}, [s,f] \in \mathcal{F}, q \in \mathcal{Q}\}$, we have

$$\begin{aligned} \min \quad & \sum_{i \in \mathcal{N}} \sum_{\ell \in \mathcal{T}} \mu_{i,\ell} + \sum_{q \in \mathcal{Q}} \sum_{j \in \mathcal{N}} \sum_{[s,f] \in \mathcal{F}} \lambda_{[s,f]}^q \sigma_{[s,f] \rightarrow j}^q & (23) \\ \text{st} \quad & \sigma_{[s,f] \rightarrow j}^q \geq \sum_{i \in \mathcal{N}} \phi_i^q(S) \beta_{i,[s,f] \rightarrow j}^q - \phi_j^q(S) \sum_{\ell=s}^f \mu_{j,\ell} \quad \forall j \in \mathcal{N}, [s,f] \in \mathcal{F}, S \subseteq \mathcal{N}, q \in \mathcal{Q} \\ & \sum_{j \in \mathcal{N}} \beta_{i,[s,f] \rightarrow j}^q = r_{i,[s,f]} \quad \forall i \in \mathcal{N}, [s,f] \in \mathcal{F} \\ & \boldsymbol{\mu} \geq \mathbf{0}, \boldsymbol{\sigma}, \boldsymbol{\beta} \text{ free.} \end{aligned}$$

It is simple to check that problem (15) is feasible and bounded, so by strong duality, problem (23) also has the optimal objective value Z_{LP}^* . We use problem (23) to give a proof for Theorem 7.2.

Proof of Theorem 7.2:

Let $(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\sigma}}, \hat{\boldsymbol{\beta}})$ be an optimal solution to problem (23). For notational brevity, also letting $\hat{\alpha}_j^q = \sum_{k=q}^Q \sum_{[s,f] \in \mathcal{F}} \lambda_{[s,f]}^k \hat{\sigma}_{[s,f] \rightarrow j}^k$, we will use induction over the time periods to establish that $V_{\hat{\boldsymbol{\beta}},j}^q(\mathbf{x}_j) \leq \hat{\alpha}_j^q + \sum_{\ell \in \mathcal{T}} \hat{\mu}_{j,\ell} x_{j,\ell}$ for all $\mathbf{x}_j \in \{0, 1\}^T$, $j \in \mathcal{N}$ and $q \in \mathcal{Q}$. In this case, using this result with $q = 1$ and $\mathbf{x}_j = \mathbf{e}'$, we obtain $\sum_{j \in \mathcal{N}} V_{\hat{\boldsymbol{\beta}},j}^1(\mathbf{e}') \leq \sum_{j \in \mathcal{N}} \hat{\alpha}_j^1 + \sum_{j \in \mathcal{N}} \sum_{\ell \in \mathcal{T}} \hat{\mu}_{j,\ell} = Z_{\text{LP}}^*$, where the equality uses the fact that $\sum_{j \in \mathcal{N}} \hat{\alpha}_j^1 + \sum_{j \in \mathcal{N}} \sum_{\ell \in \mathcal{T}} \hat{\mu}_{j,\ell} = \sum_{j \in \mathcal{N}} \sum_{q \in \mathcal{Q}} \sum_{[s,f] \in \mathcal{F}} \lambda_{[s,f]}^q \hat{\sigma}_{[s,f] \rightarrow j}^q + \sum_{j \in \mathcal{N}} \sum_{\ell \in \mathcal{T}} \hat{\mu}_{j,\ell}$ and the last quantity is the optimal objective value of problem (23), which is Z_{LP}^* . Thus, the desired result follows. In the rest of the proof, we focus on the induction argument. Noting that $\hat{\boldsymbol{\mu}} \geq \mathbf{0}$, for each $\mathbf{x}_j \in \{0, 1\}^T$, at time period $Q + 1$, we have $V_{\hat{\boldsymbol{\beta}},j}^{Q+1}(\mathbf{x}_j) = 0 \leq \sum_{\ell \in \mathcal{T}} \hat{\mu}_{j,\ell} x_{j,\ell}$, so the result holds at time period $Q + 1$. Assuming that the result holds at time period $q + 1$, we show that the result holds at time period q as well. By the induction hypothesis $V_{\hat{\boldsymbol{\beta}},j}^{q+1}(\mathbf{x}_j)$ and $V_{\hat{\boldsymbol{\beta}},j}^{q+1}(\mathbf{x}_j - \mathbf{e}_{[s,f]})$ are upper bounded by $\hat{\alpha}_j^{q+1} + \sum_{\ell \in \mathcal{T}} \hat{\mu}_{j,\ell} x_{j,\ell}$ and $\hat{\alpha}_j^{q+1} + \sum_{\ell \in \mathcal{T}} \hat{\mu}_{j,\ell} x_{j,\ell} - \sum_{\ell=s}^f \hat{\mu}_{j,\ell}$. So, by (14),

$$\begin{aligned} V_{\hat{\boldsymbol{\beta}},j}^q(\mathbf{x}_j) \leq \sum_{[s,f] \in \mathcal{F}} \lambda_{[s,f]}^q \max_{S \subseteq \mathcal{N}} \left\{ \phi_j^q(S) \left(\prod_{\ell=s}^f x_{j,\ell} \right) \left[\beta_{j,[s,f] \rightarrow j}^q - \sum_{\ell=s}^f \hat{\mu}_{j,\ell} \right] + \sum_{i \in \mathcal{N} \setminus \{j\}} \phi_i^q(S) \beta_{i,[s,f] \rightarrow j}^q \right\} \\ + \hat{\alpha}_j^{q+1} + \sum_{\ell \in \mathcal{T}} \hat{\mu}_{j,\ell} x_{j,\ell}. \end{aligned}$$

Following the same argument that we used to obtain the inequality (a) in the proof of Proposition 5.4, we can drop the product $\prod_{\ell=s}^f x_{j,\ell}$ on the right side above to make the right side of

the inequality even larger. After dropping the product $\prod_{\ell=s}^f x_{j,\ell}$, let $S_{[s,f] \rightarrow j}^*$ be an optimal solution to the resulting maximization problem. In this case, using the inequality above, we get

$$\begin{aligned}
V_{\hat{\beta},j}^q(\mathbf{x}_j) &\leq \sum_{[s,f] \in \mathcal{F}} \lambda_{[s,f]}^q \max_{S \subseteq \mathcal{N}} \left\{ \phi_j^q(S) \left[\beta_{j,[s,f] \rightarrow j}^q - \sum_{\ell=s}^f \hat{\mu}_{j,\ell} \right] + \sum_{i \in \mathcal{N} \setminus \{j\}} \phi_i^q(S) \beta_{i,[s,f] \rightarrow j}^q \right\} \\
&\quad + \hat{\alpha}_j^{q+1} + \sum_{\ell \in \mathcal{T}} \hat{\mu}_{j,\ell} x_{j,\ell} \\
&\stackrel{(a)}{=} \sum_{[s,f] \in \mathcal{F}} \lambda_{[s,f]}^q \left\{ \phi_j^q(S_{[s,f] \rightarrow j}^*) \left[\beta_{j,[s,f] \rightarrow j}^q - \sum_{\ell=s}^f \hat{\mu}_{j,\ell} \right] + \sum_{i \in \mathcal{N} \setminus \{j\}} \phi_i^q(S_{[s,f] \rightarrow j}^*) \beta_{i,[s,f] \rightarrow j}^q \right\} + \hat{\alpha}_j^{q+1} + \sum_{\ell \in \mathcal{T}} \hat{\mu}_{j,\ell} x_{j,\ell} \\
&\stackrel{(b)}{=} \sum_{[s,f] \in \mathcal{F}} \lambda_{[s,f]}^q \left\{ \sum_{i \in \mathcal{N}} \phi_i^q(S_{[s,f] \rightarrow j}^*) \beta_{i,[s,f] \rightarrow j}^q - \phi_j^q(S_{[s,f] \rightarrow j}^*) \sum_{\ell=s}^f \hat{\mu}_{j,\ell} \right\} + \hat{\alpha}_j^{q+1} + \sum_{\ell \in \mathcal{T}} \hat{\mu}_{j,\ell} x_{j,\ell} \\
&\stackrel{(c)}{\leq} \sum_{[s,f] \in \mathcal{F}} \lambda_{[s,f]}^q \hat{\sigma}_{[s,f] \rightarrow j}^q + \hat{\alpha}_j^{q+1} + \sum_{\ell \in \mathcal{T}} \hat{\mu}_{j,\ell} x_{j,\ell} \stackrel{(d)}{=} \hat{\alpha}_j^q + \sum_{\ell \in \mathcal{T}} \hat{\mu}_{j,\ell} x_{j,\ell},
\end{aligned}$$

where (a) holds because $S_{[s,f] \rightarrow j}^*$ is an optimal solution to the maximization problem on the left side of (a), (b) follows by arranging terms, (c) holds because $(\hat{\mu}, \hat{\sigma}, \hat{\beta})$ satisfies the first constraint in (23), and (d) follows because $\hat{\alpha}_j^q = \hat{\alpha}_j^{q+1} + \sum_{[s,f] \in \mathcal{F}} \lambda_{[s,f]}^q \hat{\sigma}_{[s,f] \rightarrow j}^q$. So, the induction is complete. ■

We focus on the proof of Theorem 7.3. To give an alternative representation of the value functions $\{\Gamma_{\beta,j}^q : q \in \mathcal{Q}\}$ in (16), for each $\ell \in \mathcal{T}$, define the value function

$$\Psi_{\beta,j}^q(\ell) = \sum_{k=q}^Q \sum_{[s,f] \in \mathcal{F}} \mathbf{1}_{\{s=\ell\}} \lambda_{[s,f]}^k \max_{S \subseteq \mathcal{N} \setminus \{j\}} \left\{ \sum_{i \in \mathcal{N} \setminus \{j\}} \phi_i^k(S) \beta_{i,[s,f] \rightarrow j}^k \right\}. \quad (24)$$

Directly by comparing (24) with (16), observe that the value functions $\{\Gamma_{\beta,j}^q : q \in \mathcal{Q}\}$ and $\{\Psi_{\beta,j}^q : q \in \mathcal{Q}\}$ satisfy the relationship $\Gamma_{\beta,j}^q(a,b) = \sum_{\ell=a}^b \Psi_{\beta,j}^q(\ell)$ for each interval $[a,b]$. Given that the state of resource j is $\mathbf{x}_j \in \{0,1\}^T$, let $\mathcal{K}(\mathbf{x}_j)$ be the set of unavailable days for this resource; that is $\ell \in \mathcal{K}(\mathbf{x}_j)$ if and only if $x_{j,\ell} = 0$. Note that $\mathcal{K}(\mathbf{x}_j)$ is the union of the maximal unavailable intervals with respect to \mathbf{x}_j . In other words, we have $\mathcal{K}(\mathbf{x}_j) = \cup_{[a,b] \in \mathcal{H}(\mathbf{x}_j)} [a,b]$. In this case, by the discussion earlier in this paragraph, we obtain the identity

$$\sum_{[a,b] \in \mathcal{H}(\mathbf{x}_j)} \Gamma_{\beta,j}^q(a,b) = \sum_{[a,b] \in \mathcal{H}(\mathbf{x}_j)} \sum_{\ell \in [a,b]} \Psi_{\beta,j}^q(\ell) = \sum_{\ell \in \mathcal{K}(\mathbf{x}_j)} \Psi_{\beta,j}^q(\ell),$$

where the last equality holds since $\mathcal{K}(\mathbf{x}_j) = \cup_{[a,b] \in \mathcal{H}(\mathbf{x}_j)} [a,b]$. Thus, to establish Theorem 7.3, it is enough to show that $V_{\beta,j}^q(\mathbf{x}_j) = \sum_{[a,b] \in \mathcal{I}(\mathbf{x}_j)} \Theta_{\beta,j}^q(a,b) + \sum_{\ell \in \mathcal{K}(\mathbf{x}_j)} \Psi_{\beta,j}^q(\ell)$.

Proof of Theorem 7.3:

In the proof, we will use an induction argument over the time periods to establish that $V_{\beta,j}^q(\mathbf{x}_j) = \sum_{[a,b] \in \mathcal{I}(\mathbf{x}_j)} \Theta_{\beta,j}^q(a,b) + \sum_{\ell \in \mathcal{K}(\mathbf{x}_j)} \Psi_{\beta,j}^q(\ell)$ for all $\mathbf{x}_j \in \{0,1\}^T$ and $q \in \mathcal{Q}$. At time period

$Q + 1$, we have $V_{\beta,j}^{Q+1} = \Theta_{\beta,j}^{Q+1} = \Psi_{\beta,j}^{Q+1} = 0$, so the result holds at time period $Q + 1$. Assuming that the result holds at time period $q + 1$, we show that the result holds at time period q as well. For each $\mathbf{x}_j \in \{0, 1\}^T$, the collection of maximal available intervals $\mathcal{I}(\mathbf{x}_j)$ and the set of unavailable days $\mathcal{K}(\mathbf{x}_j)$ collectively cover \mathcal{T} ; that is, $\mathcal{T} = (\cup_{[a,b] \in \mathcal{I}(\mathbf{x}_j)} [a, b]) \cup \mathcal{K}(\mathbf{x}_j)$. Thus, for each $s \in \mathcal{T}$, we have $\sum_{[a,b] \in \mathcal{I}(\mathbf{x}_j)} \mathbb{1}_{\{s \in [a,b]\}} + \sum_{\ell \in \mathcal{K}(\mathbf{x}_j)} \mathbb{1}_{\{s=\ell\}} = 1$. In this case, by (14), we get

$$\begin{aligned}
V_{\beta,j}^q(\mathbf{x}_j) &= V_{\beta,j}^{q+1}(\mathbf{x}_j) + \sum_{[s,f] \in \mathcal{F}} \lambda_{[s,f]}^q \left(\sum_{[a,b] \in \mathcal{I}(\mathbf{x}_j)} \mathbb{1}_{\{s \in [a,b]\}} + \sum_{\ell \in \mathcal{K}(\mathbf{x}_j)} \mathbb{1}_{\{s=\ell\}} \right) \times \\
&\quad \max_{S \subseteq \mathcal{N}} \left\{ \phi_j^q(S) \left(\prod_{h=s}^f x_{j,h} \right) \left[\beta_{j,[s,f] \rightarrow j}^q + V_{\beta,j}^{q+1}(\mathbf{x}_j - \mathbf{e}_{[s,f]}) - V_{\beta,j}^{q+1}(\mathbf{x}_j) \right] + \sum_{i \in \mathcal{N} \setminus \{j\}} \phi_i^q(S) \beta_{i,[s,f] \rightarrow j}^q \right\} \\
&= \sum_{[a,b] \in \mathcal{I}(\mathbf{x}_j)} \left(\Theta_{\beta,j}^{q+1}(a, b) + \sum_{[s,f] \in \mathcal{F}} \mathbb{1}_{\{s \in [a,b]\}} \lambda_{[s,f]}^q \times \right. \\
&\quad \left. \max_{S \subseteq \mathcal{N}} \left\{ \phi_j^q(S) \left(\prod_{h=s}^f x_{j,h} \right) \left[\beta_{j,[s,f] \rightarrow j}^q + V_{\beta,j}^{q+1}(\mathbf{x}_j - \mathbf{e}_{[s,f]}) - V_{\beta,j}^{q+1}(\mathbf{x}_j) \right] + \sum_{i \in \mathcal{N} \setminus \{j\}} \phi_i^q(S) \beta_{i,[s,f] \rightarrow j}^q \right\} \right) \\
&+ \sum_{\ell \in \mathcal{K}(\mathbf{x}_j)} \left(\Psi_{\beta,j}^{q+1}(\ell) + \sum_{[s,f] \in \mathcal{F}} \mathbb{1}_{\{s=\ell\}} \lambda_{[s,f]}^q \times \right. \\
&\quad \left. \max_{S \subseteq \mathcal{N}} \left\{ \phi_j^q(S) \left(\prod_{h=s}^f x_{j,h} \right) \left[\beta_{j,[s,f] \rightarrow j}^q + V_{\beta,j}^{q+1}(\mathbf{x}_j - \mathbf{e}_{[s,f]}) - V_{\beta,j}^{q+1}(\mathbf{x}_j) \right] + \sum_{i \in \mathcal{N} \setminus \{j\}} \phi_i^q(S) \beta_{i,[s,f] \rightarrow j}^q \right\} \right),
\end{aligned}$$

where the second equality follows by using the induction hypothesis to note that $V_{\beta,j}^{q+1}(\mathbf{x}_j) = \sum_{[a,b] \in \mathcal{I}(\mathbf{x}_j)} \Theta_{\beta,j}^{q+1}(a, b) + \sum_{\ell \in \mathcal{K}(\mathbf{x}_j)} \Psi_{\beta,j}^{q+1}(\ell)$ and rearranging the order of sums.

Noting the right side of the chain of equalities above, in the rest of the proof, for each $[a, b] \in \mathcal{I}(\mathbf{x}_j)$ and $\ell \in \mathcal{K}(\mathbf{x}_j)$, we will establish the following two equalities

$$\begin{aligned}
\Theta_{\beta,j}^q(a, b) &= \Theta_{\beta,j}^{q+1}([a, b]) + \sum_{[s,f] \in \mathcal{F}} \mathbb{1}_{\{s \in [a,b]\}} \lambda_{[s,f]}^q \times \\
&\quad \max_{S \subseteq \mathcal{N}} \left\{ \phi_j^q(S) \left(\prod_{h=s}^f x_{j,h} \right) \left[\beta_{j,[s,f] \rightarrow j}^q + V_{\beta,j}^{q+1}(\mathbf{x}_j - \mathbf{e}_{[s,f]}) - V_{\beta,j}^{q+1}(\mathbf{x}_j) \right] + \sum_{i \in \mathcal{N} \setminus \{j\}} \phi_i^q(S) \beta_{i,[s,f] \rightarrow j}^q \right\}, \quad (25)
\end{aligned}$$

$$\begin{aligned}
\Psi_{\beta,j}^q(\ell) &= \Psi_{\beta,j}^{q+1}(\ell) + \sum_{[s,f] \in \mathcal{F}} \mathbb{1}_{\{s=\ell\}} \lambda_{[s,f]}^q \times \\
&\quad \max_{S \subseteq \mathcal{N}} \left\{ \phi_j^q(S) \left(\prod_{h=s}^f x_{j,h} \right) \left[\beta_{j,[s,f] \rightarrow j}^q + V_{\beta,j}^{q+1}(\mathbf{x}_j - \mathbf{e}_{[s,f]}) - V_{\beta,j}^{q+1}(\mathbf{x}_j) \right] + \sum_{i \in \mathcal{N} \setminus \{j\}} \phi_i^q(S) \beta_{i,[s,f] \rightarrow j}^q \right\}. \quad (26)
\end{aligned}$$

Thus, by the first displayed equality in the proof, $V_{\beta,j}^q(\mathbf{x}_j) = \sum_{[a,b] \in \mathcal{I}(\mathbf{x}_j)} \Theta_{\beta,j}^q(a, b) + \sum_{\ell \in \mathcal{K}(\mathbf{x}_j)} \Psi_{\beta,j}^q(\ell)$, completing the induction argument.

First, we show that (25) holds for each $[a, b] \in \mathcal{I}(\mathbf{x}_j)$. So, throughout this portion of the discussion, we focus on the maximal available intervals $[a, b] \in \mathcal{I}(\mathbf{x}_j)$. In (25), we consider $s \in [a, b]$. Note that if

$s \in [a, b]$ and $f \in [a, b]$, then because $[a, b] \in \mathcal{I}(\mathbf{x}_j)$, we get $x_{j,h} = 1$ for all $h = s, \dots, f$, so $\prod_{h=s}^f x_{j,h} = 1$. Furthermore, because $s \in [a, b]$, we have $f \in [a, b]$ if and only if $[s, f] \subseteq [a, b]$. On the other hand, if $s \in [a, b]$ and $f \notin [a, b]$, then since the interval $[a, b]$ is a maximal available interval, there must be some $h = s + 1, \dots, f$ such that $x_{j,h} = 0$, so $\prod_{h=s}^f x_{j,h} = 0$. Furthermore, because $s \in [a, b]$, we have $f \notin [a, b]$ if and only if $[s, f] \not\subseteq [a, b]$. Thus, using $\mathbf{1}_{\{s \in [a, b]\}} = \mathbf{1}_{\{s \in [a, b], [s, f] \subseteq [a, b]\}} + \mathbf{1}_{\{s \in [a, b], [s, f] \not\subseteq [a, b]\}}$, we equivalently express the right side of (25) as

$$\begin{aligned}
& \Theta_{\beta,j}^{q+1}(a, b) + \sum_{[s,f] \in \mathcal{F}} \mathbf{1}_{\{s \in [a, b], [s, f] \subseteq [a, b]\}} \lambda_{[s,f]}^q \times \\
& \quad \max_{S \subseteq \mathcal{N}} \left\{ \phi_j^q(S) \left(\prod_{h=s}^f x_{j,h} \right) \left[\beta_{j,[s,f] \rightarrow j}^q + V_{\beta,j}^{q+1}(\mathbf{x}_j - \mathbf{e}_{[s,f]}) - V_{\beta,j}^{q+1}(\mathbf{x}_j) \right] + \sum_{i \in \mathcal{N} \setminus \{j\}} \phi_i^q(S) \beta_{i,[s,f] \rightarrow j}^q \right\} \\
& \quad + \sum_{[s,f] \in \mathcal{F}} \mathbf{1}_{\{s \in [a, b], [s, f] \not\subseteq [a, b]\}} \lambda_{[s,f]}^q \times \\
& \quad \max_{S \subseteq \mathcal{N}} \left\{ \phi_j^q(S) \left(\prod_{h=s}^f x_{j,h} \right) \left[\beta_{j,[s,f] \rightarrow j}^q + V_{\beta,j}^{q+1}(\mathbf{x}_j - \mathbf{e}_{[s,f]}) - V_{\beta,j}^{q+1}(\mathbf{x}_j) \right] + \sum_{i \in \mathcal{N} \setminus \{j\}} \phi_i^q(S) \beta_{i,[s,f] \rightarrow j}^q \right\} \\
& \stackrel{(a)}{=} \Theta_{\beta,j}^{q+1}(a, b) + \sum_{[s,f] \in \mathcal{F}} \mathbf{1}_{\{s \in [a, b], [s, f] \subseteq [a, b]\}} \lambda_{[s,f]}^q \times \\
& \quad \max_{S \subseteq \mathcal{N}} \left\{ \phi_j^q(S) \left[\beta_{j,[s,f] \rightarrow j}^q + V_{\beta,j}^{q+1}(\mathbf{x}_j - \mathbf{e}_{[s,f]}) - V_{\beta,j}^{q+1}(\mathbf{x}_j) \right] + \sum_{i \in \mathcal{N} \setminus \{j\}} \phi_i^q(S) \beta_{i,[s,f] \rightarrow j}^q \right\} \\
& \quad + \sum_{[s,f] \in \mathcal{F}} \mathbf{1}_{\{s \in [a, b], [s, f] \not\subseteq [a, b]\}} \lambda_{[s,f]}^q \max_{S \subseteq \mathcal{N} \setminus \{j\}} \left\{ \sum_{i \in \mathcal{N} \setminus \{j\}} \phi_i^q(S) \beta_{i,[s,f] \rightarrow j}^q \right\}, \tag{27}
\end{aligned}$$

where (a) holds because if $[s, f] \subseteq [a, b]$, then $\prod_{h=s}^f x_{j,h} = 1$, whereas if $s \in [a, b]$ and $[s, f] \not\subseteq [a, b]$, then $\prod_{h=s}^f x_{j,h} = 0$, as discussed right before the chain of equalities just above.

From Section 3, recall that $\mathcal{I}(\mathbf{x}_j - \mathbf{e}_{[s,f]}) = (\mathcal{I}(\mathbf{x}_j) \setminus [a, b]) \cup \{[a, s-1], [f+1, b]\}$ for $[s, f] \subseteq [a, b]$. By definition of $\mathcal{K}(\mathbf{x}_j)$, $\mathcal{K}(\mathbf{x}_j - \mathbf{e}_{[s,f]}) = \mathcal{K}(\mathbf{x}_j) \cup \{s, \dots, f\}$. So, by the induction hypothesis, we get

$$\begin{aligned}
& V_{\beta,j}^{q+1}(\mathbf{x}_j - \mathbf{e}_{[s,f]}) - V_{\beta,j}^{q+1}(\mathbf{x}_j) \\
& = \sum_{[a,b] \in \mathcal{I}(\mathbf{x}_j - \mathbf{e}_{[s,f]})} \Theta_{\beta,j}^{q+1}(a, b) + \sum_{\ell \in \mathcal{K}(\mathbf{x}_j - \mathbf{e}_{[s,f]})} \Psi_{\beta,j}^{q+1}(\ell) - \sum_{[a,b] \in \mathcal{I}(\mathbf{x}_j)} \Theta_{\beta,j}^{q+1}(a, b) - \sum_{\ell \in \mathcal{K}(\mathbf{x}_j)} \Psi_{\beta,j}^{q+1}(\ell) \\
& = \Theta_{\beta,j}^{q+1}(a, s-1) + \Theta_{\beta,j}^{q+1}(f+1, b) - \Theta_{\beta,j}^{q+1}(a, b) + \sum_{\ell=s}^f \Psi_{\beta,j}^{q+1}(\ell) \\
& \stackrel{(b)}{=} \Theta_{\beta,j}^{q+1}(a, s-1) + \Theta_{\beta,j}^{q+1}(f+1, b) - \Theta_{\beta,j}^{q+1}(a, b) + \Gamma_{\beta,j}^{q+1}(s, f), \tag{28}
\end{aligned}$$

where (b) follows from the discussion right before the proof of the theorem, which shows that the value functions $\{\Gamma_{\beta,j}^q : q \in \mathcal{Q}\}$ and $\{\Psi_{\beta,j}^q : q \in \mathcal{Q}\}$ computed, respectively, through (16) and (24)

satisfy the identity $\Gamma_{\beta,j}^{q+1}(s, f) = \sum_{\ell=s}^f \Psi_{\beta,j}^{q+1}(\ell)$. In this case, plugging (28) into (27), we equivalently express the right side of (25) as

$$\begin{aligned} & \Theta_{\beta,j}^{q+1}(a, b) + \sum_{[s,f] \in \mathcal{F}} \mathbf{1}_{\{s \in [a,b], [s,f] \subseteq [a,b]\}} \lambda_{[s,f]}^q \times \\ & \quad \max_{S \subseteq \mathcal{N}} \left\{ \phi_j^q(S) \left[\beta_{j,[s,f] \rightarrow j}^q + \Theta_{\beta,j}^{q+1}(a, s-1) + \Theta_{\beta,j}^{q+1}(f+1, b) - \Theta_{\beta,j}^{q+1}(a, b) + \Gamma_{\beta,j}^{q+1}(s, f) \right] \right. \\ & \quad \quad \quad \left. + \sum_{i \in \mathcal{N} \setminus \{j\}} \phi_i^q(S) \beta_{i,[s,f] \rightarrow j}^q \right\} \\ & \quad + \sum_{[s,f] \in \mathcal{F}} \mathbf{1}_{\{s \in [a,b], [s,f] \not\subseteq [a,b]\}} \lambda_{[s,f]}^q \max_{S \subseteq \mathcal{N} \setminus \{j\}} \left\{ \sum_{i \in \mathcal{N} \setminus \{j\}} \phi_i^q(S) \beta_{i,[s,f] \rightarrow j}^q \right\} \\ & \stackrel{(c)}{=} \Theta_{\beta,j}^q(a, b), \end{aligned}$$

where (c) follows from (17). By the equality above, the right side of (25) is equal to $\Theta_{\beta,j}^q(a, b)$, establishing the equality in (25).

Second, we show that (26) holds for each $\ell \in \mathcal{K}(\mathbf{x}_j)$. Thus, we focus on the unavailable days $\ell \in \mathcal{K}(\mathbf{x}_j)$. For $\ell \in \mathcal{K}(\mathbf{x}_j)$, we have $x_{j,\ell} = 0$, so $\prod_{h=s}^f x_{j,h} = 0$ for $s = \ell$. So, the right side of (26) reads

$$\begin{aligned} & \Psi_{\beta,j}^{q+1}(\ell) + \sum_{[s,f] \in \mathcal{F}} \mathbf{1}_{\{s=\ell\}} \lambda_{[s,f]}^q \times \\ & \quad \max_{S \subseteq \mathcal{N}} \left\{ \phi_j^q(S) \left(\prod_{h=s}^f x_{j,h} \right) \left[\beta_{j,[s,f] \rightarrow j}^q + V_{\beta,j}^{q+1}(\mathbf{x}_j - \mathbf{e}_{[s,f]}) - V_{\beta,j}^{q+1}(\mathbf{x}_j) \right] + \sum_{i \in \mathcal{N} \setminus \{j\}} \phi_i^q(S) \beta_{i,[s,f] \rightarrow j}^q \right\} \\ & = \Psi_{\beta,j}^{q+1}(\ell) + \sum_{[s,f] \in \mathcal{F}} \mathbf{1}_{\{s=\ell\}} \lambda_{[s,f]}^q \max_{S \subseteq \mathcal{N}} \left\{ \sum_{i \in \mathcal{N} \setminus \{j\}} \phi_i^q(S) \beta_{i,[s,f] \rightarrow j}^q \right\} \\ & \stackrel{(d)}{=} \Psi_{\beta,j}^{q+1}(\ell) + \sum_{[s,f] \in \mathcal{F}} \mathbf{1}_{\{s=\ell\}} \lambda_{[s,f]}^q \max_{S \subseteq \mathcal{N} \setminus \{j\}} \left\{ \sum_{i \in \mathcal{N} \setminus \{j\}} \phi_i^q(S) \beta_{i,[s,f] \rightarrow j}^q \right\} \stackrel{(e)}{=} \Psi_{\beta,j}^q(\ell), \end{aligned}$$

where (d) holds because resource j does not appear in the objective function of the maximization problem on the left side of (d), and (e) follows by (24). Thus, the equality in (26) holds. \blacksquare

Appendix E: Performance of the Constraint Splitting Policy on Synthetic Datasets

In Baek and Ma (2019), the authors consider general revenue management problems with non-unit resource capacities, as well as booking requests not necessarily over intervals of days. They split the resource constraints into two groups. In the first group, a booking request consumes the capacities of at most L different resources. In the second group, they have a matroid characterization of the capacity consumptions of the booking requests. The authors give a policy that is guaranteed to obtain at least $\frac{1}{2(1+L)}$ fraction of the optimal total expected revenue. We

Param. (D_{\max}, ρ, δ)	Total Exp. Revenue				
	LINR	POLR	LPR	COS1	COS5
(6, 1.2, 0.9)	81.05	80.29	81.11	65.40	63.16
(6, 1.2, 0.7)	73.00	73.04	73.97	64.92	61.33
(6, 1.6, 0.9)	78.60	75.96	77.92	67.19	63.48
(6, 1.6, 0.7)	85.02	83.07	84.52	74.50	70.95
(6, 2.0, 0.9)	85.04	82.84	84.14	77.08	72.50
(6, 2.0, 0.7)	80.85	77.39	79.28	73.25	69.05
(8, 1.2, 0.9)	78.23	77.84	79.29	62.82	61.58
(8, 1.2, 0.7)	71.07	71.09	73.44	59.86	56.76
(8, 1.6, 0.9)	84.64	81.96	84.82	70.59	69.37
(8, 1.6, 0.7)	79.75	78.52	81.57	66.73	63.04
(8, 2.0, 0.9)	84.26	81.37	81.56	69.90	67.93
(8, 2.0, 0.7)	79.21	73.62	77.22	65.19	62.08
(10, 1.2, 0.9)	73.89	71.43	74.33	53.93	54.13
(10, 1.2, 0.7)	73.09	72.90	75.02	57.47	58.17
(10, 1.6, 0.9)	79.11	74.98	78.40	60.83	60.13
(10, 1.6, 0.7)	82.22	78.10	81.86	62.89	59.45
(10, 2.0, 0.9)	83.65	79.14	81.83	67.78	64.75
(10, 2.0, 0.7)	83.93	75.64	82.01	67.59	59.69
Avg.	79.81	77.18	79.57	66.00	63.20

Table EC.1 Total expected revenues obtained by the constraint splitting policies.

refer to the policy proposed by Baek and Ma (2019) as the constraint splitting policy. The performance guarantee for the constraint splitting policy applies to general revenue management problems, not requiring the resources to be unique or the booking requests to consume capacities on consecutive days. Furthermore, this performance guarantee is independent of the number of constraints in the second group. The constraint splitting policy, when applied to our revenue management problem, provides a performance guarantee of $\frac{1}{1+D_{\max}}$. Our polynomial value function approximations improve upon this performance guarantee when $D_{\min} > 1$. In this paper, exploiting the unit capacities of the resources and the interval structure of the booking requests, we can further perform rollout on static policies, which dramatically improves the performance of the static policy on hand. For our synthetic datasets, we compute the parameters of the constraint splitting policy once or five times over the selling horizon. We refer to the resulting policies, respectively, as COS1 and COS5, where we emphasize that these policies are based on constraint splitting. In Table EC.1, we compare the total expected revenues obtained by COS1 and COS5 with those obtained by LINR, POLR, and LPR. We express the total expected revenue of each benchmark policy as a percentage of the upper bound on the optimal total expected revenue. The total expected revenues that we report for LINR, POLR, and LPR in Table EC.1 are taken from Table 2.

Our results indicate that LINR, POLR, and LPR perform significantly better than COS1 and COS5. On average, the total expected revenues obtained by COS5 lag behind those obtained by LINR, POLR, and LPR by, respectively, 20.83%, 18.19%, and 20.63%. Thus, exploiting the structure of our revenue management problem, our rollout policies provide substantial improvements over COS5. Similar observations hold for COS1 as well. Comparing the performance of COS5 with that

of LIN5, POL5, and LP5 from Table 2, the improvements of LIN5, POL5, and LP5 over COS5 are still significant, reaching, respectively, 19.01%, 7.49%, and 16.69%, on average.

Appendix F: Additional Details for the Boutique Hotel Dataset

We explain the details of the parameter estimates for the boutique hotel dataset. Following this discussion, we check the performance of the policy in Baek and Ma (2019).

Parameter Estimation: Recalling that the values of $\{\theta_k : k = 1, \dots, 171\}$ are the same when k falls in each of the seven intervals $[1, 3]$, $[4, 7]$, $[8, 14]$, $[15, 28]$, $[29, 42]$, $[43, 56]$ and $[57, 171]$, we need to estimate seven parameters to come up with $\{\theta_k : k = 1, \dots, 171\}$. The length of stay for a customer ranges between one and eight days, so we need to estimate eight parameters for $\{\eta_d : d = 1, \dots, 8\}$. Lastly, there are six rooms, which implies that we need to estimate the six preference weights $\{v_i : i = 1, \dots, 6\}$. Thus, the total number of parameters that we need to estimate is 21. For each one of the time periods in the selling horizon, the dataset provides the assortment of rooms that was on offer to the customers and indicates whether there was a booking at the time period. If there was a booking, then the dataset also provides the room chosen in the booking and the interval of stay for the booking. Using the dataset, we use standard maximum likelihood estimation to estimate the parameters of our model. We provide summary statistics for the estimated values of the parameters. The largest values for θ_k occur when k falls in one of the intervals $[1, 3]$, $[4, 7]$ and $[8, 14]$. More than 60% of the booking requests are for intervals of three or fewer days. Lastly, the largest and smallest preference weights for a room differ by a factor of 2.28.

We carry out five-fold cross-validation for our arrival probability and preference weight estimates. Recall that we have 1190 time periods in our selling horizon. Some time periods have bookings, some do not. We split the 1190 time periods in the selling horizon into five equal segments. After estimating the parameters of our model using four-fifths of the dataset, we validate the ability of our parameter estimates to predict the arrivals and customer choices in the remaining one-fifth holdout dataset. To validate the estimated arrival probabilities and preference weights, we use our model parameters to predict the expected number of weekly bookings made within the assortments offered in the holdout dataset, and compare our predictions with the actual numbers of bookings. Over five holdout datasets, the average percent deviation between the predicted and actual bookings is 23%. For each time period with a reservation in the holdout dataset, we also order the offered rooms according to their choice probabilities from the fitted choice model, and count the fraction of times that the booked room is one of the r rooms with the largest choice probabilities. We refer to this fraction as the r -hit rate. For example, the 2-hit rate is the fraction of times that the booked room had one of the top two choice probabilities. The 1-hit and 2-hit

Load Fact.	Total Exp. Revenue				
	LINR	POLR	LPR	COS1	COS5
0.8	83.96	84.43	83.00	78.07	78.22
1.2	88.18	88.70	87.16	82.93	83.64
1.6	90.18	89.69	88.85	85.18	84.77
2.0	91.03	91.22	90.53	86.75	86.63
Avg.	88.34	88.51	87.38	83.23	83.32

Table EC.2 Total expected revenues obtained by the constraint splitting policies.

rates, averaged over five holdout datasets, are, respectively, 0.58 and 0.82. Therefore, more than 80% of the time, the booked room is one of the two options with the largest choice probabilities. Similarly, more than 50% of the time, the booked room is indeed the one with the largest choice probability. Lastly, the average rank of the booked room, averaged over all holdout datasets and bookings in each holdout dataset, is 1.75.

Performance of the Constraint Splitting Policy: As discussed in Appendix E, we compute the parameters of the constraint splitting policy in Baek and Ma (2019) once and five times over the selling horizon, yielding the policies COS1 and COS5. In Table EC.2, we compare the performance of these two policies with LINR, POLR, and LPR. Our results indicate that each of the policies LINR, POLR, and LPR provide noticeable improvements over COS1 and COS5.