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

Paat Rusmevichientong[1]*, Mika Sumida[1], Huseyin Topaloglu[2], Yicheng Bai[2]
[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 22, 2022

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 total 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 $\mathsf{D}_{\max}$ 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 $\frac{1}{2\,\mathsf{D}_{\max}}$ 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 $\mathsf{D}_{\min}$ 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 $\frac{1}{2+\lceil(\mathsf{D}_{\max}-1)/\mathsf{D}_{\min}\rceil}$ fraction of the optimal total expected revenue. To establish this result, corresponding to each possible interval of days for 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 5.1). 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 $1+\lceil(\mathsf{D}_{\max}-1)/\mathsf{D}_{\min}\rceil$ (Theorem 5.2). Putting these two results together yields the desired performance guarantee. Surprisingly, if all booking requests are for the same duration ($\mathsf{D}_{\max}=\mathsf{D}_{\min}$), 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 6.1). 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 6.2). 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 $O(T^2)$ states (Theorem 6.3).

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 the components of the state vector; see Topaloglu (2009). The approach in Topaloglu (2009) constructs value function approximations that are separable by each resource and day of use, whereas our approach constructs value function approximations that are separable by each resource, but not by day of use. In our experiments, the upper bound from Topaloglu (2009) 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. Therefore, if the initial policy on hand has a performance guarantee, then the same performance guarantee applies to the policy from rollout, but the practical performance of the rollout policy is often reported to be significantly better than that of the initial policy. The difficulty of performing rollout is in computing the value functions of the initial policy on hand. Computing the value functions of the initial policy can be as difficult as computing the optimal policy itself, so these value functions are often estimated by simulating the performance of the initial policy on hand. In contrast, we are able to 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, yielding policies stronger than the static policy. Rollout 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 with reusable features. Rusmevichientong et al. (2020) consider a problem with reusable resources where the resources are rented for random durations of time and become available to be used by other customers once returned. The authors give a policy that obtains at least 50% of the optimal total expected revenue. An important differentiating factor in 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 has advance reservations, which is, naturally, critical for lodging market places, hotels and freelancer matching platforms. As far as we are aware, it is not possible to incorporate advance reservations into the policy proposed by Rusmevichientong et al. (2020). In particular, the authors use the fact that if they have single unit of a resource, then they can find the optimal policy to manage this unit by solving a simple dynamic program. Letting T be the number of time periods in the planning horizon and K be the maximum number of days that a resource can be rented, their value functions are of the form ν^t_ℓ for $t = 1, 2, \ldots, T$, $\ell = 0, 1, \ldots, K$, where ν^t_ℓ is the optimal total expected revenue from a unit of the resource over time periods $t, \ldots, T$ given that this unit has been in use for ℓ time periods at the beginning of time period t. Since their problem does not allow advance reservations, they only need to keep track of whether the unit is in use and (if it is) for how many time periods it has been in use. They do not need to keep track of future time periods at which the resource might be committed for use. In our problem, even if we manage single unit of a resource, we cannot characterize the optimal policy in such a simple fashion. We need to keep track of the availability of the resource on each day in the future. We get around this difficulty as follows. First, under our linear approximations, we show that we can use a linear function of the capacities to check whether we can serve a booking request for a set of future days. Second, under our polynomial approximations, we show that we can use the intersection preserving subsets for the same purpose. Third, once we construct our static policies, we show that we can efficiently compute the value functions of a static policy by focusing on uninterrupted availabilities over intervals of days. All of these results are possible because the resources have unit capacities, the booking requests are for intervals of days and we focus on static policies. Rusmevichientong et al. (2020) use rollout but they can do so only when there are no advance reservations and the usage durations are negative binomial. They cannot perform rollout even when the usage durations are deterministically known at the time of booking, which is the relevant case for lodging market places, hotels and freelancer matching platforms. Manshadi and Rodilitz (2020) consider a problem with reusable resources, where volunteers are asked to perform tasks, each volunteer becoming inactive for a duration of time after she is tapped for a task. It is not clear how to incorporate advance reservations into their work either.

Considering non-unit resource capacities and booking requests not necessarily over intervals of days, letting L be the maximum number of days used by a booking request, Baek and Ma (2021) give a policy with a performance guarantee of $\frac{1}{1+L}$. Their setting is more general than ours, but exploiting the special structure of our problem, we give improvements in three directions. First, it is not possible to compute the value functions of the policy proposed by Baek and Ma (2021), so we cannot perform rollout on their policy. Rollout provides dramatic performance improvements. Performing rollout on our static policy with polynomial approximations, we can improve the total expected revenue by more than 15%. Second, the policy in Baek and Ma (2021), when applied to our problem, provides a performance guarantee of $\frac{1}{1+\mathsf{D}_{\max}}$. If $\mathsf{D}_{\max} > \mathsf{D}_{\min} > 1$, then we have $\frac{1}{2+\lceil(\mathsf{D}_{\max}-1)/\mathsf{D}_{\min}\rceil} > \frac{1}{1+\mathsf{D}_{\max}}$, so our policy with polynomial approximations has a better performance guarantee. Platforms may indeed impose minimum use requirements, so $\mathsf{D}_{\min} > 1$. For example, in addition to its six rooms, Villa Mahal offers three villas, each requiring a minimum stay of seven days. Third, Baek and Ma (2021) use the linear programming approximation mentioned earlier in this section to construct their policy. This approximation works well in an asymptotic regime where the resource capacities are large, but leveraging the linear program when each resource has a unit capacity yields a policy that does not perform as well in practice. Our policy is designed to deal with unit capacities and can improve the one in Baek and Ma (2021) by up to 20%. Similarly, Ma et al. (2020) give a policy with a performance guarantee of $\frac{1}{1+L}$, but their policy also does not allow rollout, is insensitive to $\mathsf{D}_{\min}$ and is not designed to deal with unit resource capacities.

Some papers build 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). These papers focus on the asymptotic regime discussed earlier in this section, but such a regime, once again, is not relevant to us 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 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). These papers do not give performance guarantees either.

In our problem setting, the revenues from the booking requests are fixed and the only decision is the assortment of resources offered in response to each booking request, but all of our results hold with little modification when the prices are also decision variables. In particular, we can create multiple copies of a resource, each copy corresponding to offering the resource at a different price level. In this case, we offer an assortment of resource and price pairs in response to each booking request, naturally with the constraint that a resource, if offered, can be offered at one price level. This approach immediately extends our results to deal with a finite number of possible price levels, but we can rederive our results to deal with prices that lie on a continuum as well.

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 give a static policy using polynomial value function approximations. In Section 6, we give a method to obtain an upper bound on the optimal total expected revenue. In Section 7, we give computational experiments. In our computational experiments, we test the performance of our static and rollout policies, as well as a policy that builds on the value function approximations used to construct our upper bound on the optimal total expected revenue. In Section 8, 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 $\mathcal{F}=\{[s,f]:1\le s\le f\le T\}$ 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 $\lambda^q_{[s,f]}$. With probability $1-\sum_{[s,f]\in\mathcal{F}}\lambda^q_{[s,f]}$, 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 $\phi^q_i(S)$. The choice probability $\phi^q_i(S)$ 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, $\phi^q_i(S\cup\{j\})\le\phi^q_i(S)$ 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 $\boldsymbol{x}=(x_{i,\ell}:i\in\mathcal{N},\ \ell\in\mathcal{T})\in\{0,1\}^{N\times T}$ 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 $\boldsymbol{x}$, 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 $J^q(\boldsymbol{x})$ be the optimal total expected revenue over time periods $\{q,\dots,Q\}$ given that the state of the system at time period q is $\boldsymbol{x}$. Using $\boldsymbol{e}_{i,[s,f]}\in\{0,1\}^{N\times T}$ 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

$$
\begin{aligned}
J^q(\boldsymbol{x}) \;=\; \sum_{[s,f]\in\mathcal{F}} \lambda^q_{[s,f]} \max_{S\subseteq\mathcal{N}} \Bigg\{ & \sum_{i\in\mathcal{N}} \phi^q_i(S)\Bigg(\prod_{\ell=s}^{f} x_{i,\ell}\Bigg)\Big[r_{i,[s,f]}+J^{q+1}(\boldsymbol{x}-\boldsymbol{e}_{i,[s,f]})\Big] \\
&+\Bigg[\sum_{i\in\mathcal{N}}\phi^q_i(S)\Bigg(1-\prod_{\ell=s}^{f}x_{i,\ell}\Bigg)+1-\sum_{i\in\mathcal{N}}\phi^q_i(S)\Bigg]J^{q+1}(\boldsymbol{x})\Bigg\} \;+\; \Bigg[1-\sum_{[s,f]\in\mathcal{F}}\lambda^q_{[s,f]}\Bigg]J^{q+1}(\boldsymbol{x}),
\end{aligned}
$$

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

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 $\phi^q_i(S)$. 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

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

Given that the state of the system at time period q is $\boldsymbol{x}$, we interpret $J^{q+1}(\boldsymbol{x})-J^{q+1}(\boldsymbol{x}-\boldsymbol{e}_{i,[s,f]})$ 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 $\boldsymbol{x}$ 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 $\max_{S\subseteq\mathcal{N}} \sum_{i\in\mathcal{N}} \phi_i^q(S)\; p_{i,[s,f]}^q(\boldsymbol{x})$, where $p_{i,[s,f]}^q(\boldsymbol{x}) = (\prod_{\ell=s}^{f} x_{i,\ell})\,[r_{i,[s,f]} + J^{q+1}(\boldsymbol{x}-\boldsymbol{e}_{i,[s,f]}) - J^{q+1}(\boldsymbol{x})]$. 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 $p_i^* \le 0$, then S^* does not offer resource i. This result follows from the assumption that the choice probabilities satisfy $\phi_i^q(S\cup\{j\}) \le \phi_i^q(S)$ 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 $p_{i,[s,f]}^q(\boldsymbol{x}) = 0$. 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 $\boldsymbol{x}$ 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 $\{\mu^q_{[s,f]}(S) : S \subseteq \mathcal{N},\ [s,f] \in \mathcal{F},\ q \in \mathcal{Q}\}$ 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 $\mu^q_{[s,f]}(S)$. Naturally, we require $\sum_{S \subseteq \mathcal{N}} \mu^q_{[s,f]}(S) = 1$. 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 $\psi^q_{\boldsymbol{\mu},i,[s,f]} = \sum_{S \subseteq \mathcal{N}} \mu^q_{[s,f]}(S)\ \phi^q_i(S)$, 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 $J^q_{\boldsymbol{\mu}}(\boldsymbol{x})$ be the total expected revenue obtained by the static policy $\boldsymbol{\mu}$ over time periods $\{q, \ldots, Q\}$ given that the state of the system at time period q is $\boldsymbol{x}$. We compute the value functions $\{J^q_{\boldsymbol{\mu}} : q \in \mathcal{Q}\}$ through the dynamic program

$$J^q_{\boldsymbol{\mu}}(\boldsymbol{x}) = \sum_{[s,f] \in \mathcal{F}} \lambda^q_{[s,f]} \left\{ \sum_{i \in \mathcal{N}} \psi^q_{\boldsymbol{\mu},i,[s,f]} \left(\prod_{\ell=s}^{f} x_{i,\ell} \right) \left[r_{i,[s,f]} + J^{q+1}_{\boldsymbol{\mu}}(\boldsymbol{x} - \boldsymbol{e}_{i,[s,f]}) - J^{q+1}_{\boldsymbol{\mu}}(\boldsymbol{x}) \right] \right\} + J^{q+1}_{\boldsymbol{\mu}}(\boldsymbol{x}), \tag{2}$$

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

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 $\psi^q_{\boldsymbol{\mu},i,[s,f]}$. The state variable $\boldsymbol{x}$ 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 $\psi^q_{\boldsymbol{\mu},i,[s,f]}$. 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 $\{a, \ldots, b-1\}$ and days $\{b+1, \ldots, c\}$, 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 $V^q_{\boldsymbol{\mu},i}(a,b)$ be the total expected revenue collected by the static policy $\boldsymbol{\mu}$ from resource i over

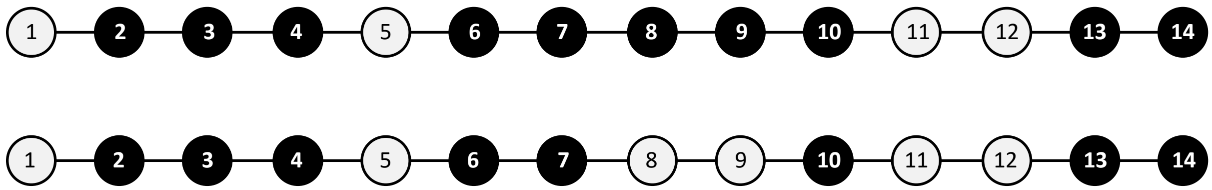

Figure 1 Maximal available intervals for $\boldsymbol{x}_i$ (top) and maximal intervals for $\boldsymbol{x}_i - \boldsymbol{e}_{[8,9]}$ (bottom).

time periods $\{q,\ldots,Q\}$ given that resource i is available for <u>*all*</u> the days in the interval $[a,b]$. We compute the value functions $\{V^q_{\boldsymbol{\mu},i} : q \in \mathcal{Q}\}$ by solving the dynamic program

$$V^q_{\boldsymbol{\mu},i}(a,b) = \sum_{[s,f]\subseteq[a,b]} \lambda^q_{[s,f]} \left\{ \psi^q_{\boldsymbol{\mu},i,[s,f]} \left[r_{i,[s,f]} + V^{q+1}_{\boldsymbol{\mu},i}(a,s-1) + V^{q+1}_{\boldsymbol{\mu},i}(f+1,b) - V^{q+1}_{\boldsymbol{\mu},i}(a,b) \right] \right\} + V^{q+1}_{\boldsymbol{\mu},i}(a,b), \quad (3)$$

with the boundary condition that $V^{Q+1}_{\boldsymbol{\mu},i} = 0$. 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 $\boldsymbol{\mu}$ from this resource over time periods $\{q,\ldots,Q\}$, we consider the booking requests only for the intervals within the interval $[a,b]$. With probability $\lambda^q_{[s,f]}$, the customer arriving at time period q makes a booking request for interval $[s,f]$. Under the static policy $\boldsymbol{\mu}$, this customer chooses resource i with probability $\psi^q_{\boldsymbol{\mu},i,[s,f]}$, 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 $V^{q+1}_{\boldsymbol{\mu},i}(a,b) - V^{q+1}_{\boldsymbol{\mu},i}(a,s-1) - V^{q+1}_{\boldsymbol{\mu},i}(f+1,b)$ in (3) as the opportunity cost of the capacities used by booking resource i for the days $\{s,\ldots,f\}$. If $s=a$, then we set $V^{q+1}_{\boldsymbol{\mu},i}(a,s-1)=0$, whereas if $f=b$, then we set $V^{q+1}_{\boldsymbol{\mu},i}(f+1,b)=0$ 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 $\{J^q_{\boldsymbol{\mu}} : q \in \mathcal{Q}\}$ in (2) by using the value functions in $\{V^q_{\boldsymbol{\mu},i} : q \in \mathcal{Q}\}$ in (3). In particular, we use $\boldsymbol{x}_i = (x_{i,1},\ldots,x_{i,T}) \in \{0,1\}^T$ 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 <u>*maximal available interval*</u> with respect to $\boldsymbol{x}_i$ 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, $\prod_{\ell=a}^b \boldsymbol{x}_{i,\ell}=1$, $x_{i,a-1}=0$ and $x_{i,b+1}=0$. Let $\mathcal{I}(x_i)$ be the collection of maximal available intervals with respect to $\boldsymbol{x}_i$. In the top portion of Figure 1, we show the collection $\mathcal{I}(\boldsymbol{x}_i)$ for a specific value of $\boldsymbol{x}_i$ 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 $\mathcal{I}(\boldsymbol{x}_i)=\{[2,4],[6,10],[13,14]\}$.

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 $[a,b]\in\mathcal{I}(\boldsymbol{x}_i)$ 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 $[a,b] \in \mathcal{I}(\boldsymbol{x}_i)$ such that $[7,9] \subseteq [a,b]$. This maximal available interval is $[6,10]$. Second, using $\boldsymbol{e}_{[s,f]} \in \{0,1\}^T$ to denote the vector with ones only in the components corresponding to the days $\{s,\ldots,f\}$, if $[a,b]$ is the maximal available interval that includes the interval $[s,f]$, then we have the identity $\mathcal{I}(\boldsymbol{x}_i - \boldsymbol{e}_{[s,f]}) = \big(\mathcal{I}(\boldsymbol{x}_i) \setminus [a,b]\big) \cup \big\{[a,s-1],\, [f+1,b]\big\}$. 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 $\mathcal{I}(\boldsymbol{x}_i) = \{[2,4],[6,10],[13,14]\}$ for the specific value of $\boldsymbol{x}_i$. 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 $\mathcal{I}(\boldsymbol{x}_i - \boldsymbol{e}_{[8,9]}) = \{[2,4],[6,7],[10,10],[13,14]\}$, which is indeed equal to $(\mathcal{I}(\boldsymbol{x}_i) \setminus [6,10]) \cup \{[6,7],[10,10]\}$.

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

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

$$J^q_{\boldsymbol{\mu}}(\boldsymbol{x}) \;=\; \sum_{i\in\mathcal{N}} \sum_{[a,b]\in\mathcal{I}(\boldsymbol{x}_i)} V^q_{\boldsymbol{\mu},i}(a,b).$$

Proof: We show the result by using induction over the time periods. At time period $Q+1$, we have $J^{Q+1}_{\boldsymbol{\mu}} = 0 = V^{Q+1}_{\boldsymbol{\mu},i}$, 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 $[a,b] \in \mathcal{I}(\boldsymbol{x}_i)$ 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 $\sum_{[a,b]\in\mathcal{I}(\boldsymbol{x}_i)} \mathbb{1}_{\{[s,f]\subseteq[a,b]\}} = 1$. 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 $\mathcal{I}(\boldsymbol{x}_i - \boldsymbol{e}_{[s,f]}) = (\mathcal{I}(\boldsymbol{x}_i) \setminus [a,b]) \cup \{[a,s-1],\, [f+1,b]\}$. In this case, considering the maximal available interval $[a,b] \in \mathcal{I}(\boldsymbol{x}_i)$, for each interval $[s,f] \subseteq [a,b]$, using the induction hypothesis, we obtain the chain of equalities

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

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

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

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

where (a) follows by reordering the three sums on the left side of (a), (b) holds because (4) holds for all $[a,b]\in\mathcal{I}(\boldsymbol{x}_i)$ 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 $\boldsymbol{x}$ and we have a booking request for interval $[s,f]$, the rollout policy from the static policy $\boldsymbol{\mu}$ offers the assortment of resources

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

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 $S_{\boldsymbol{\mu},[s,f]}^{\mathsf{Rollout},q}(\boldsymbol{x})$ to the problem above such that $i \notin S_{\boldsymbol{\mu},[s,f]}^{\mathsf{Rollout},q}(\boldsymbol{x})$ 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 $\{\hat{J}^q_{\mathsf{L}} : q \in \mathcal{Q}\}$ of the form

$$\hat{J}^q_{\mathsf{L}}(\boldsymbol{x}) \;=\; \sum_{i\in\mathcal{N}}\sum_{\ell\in\mathcal{T}} \eta^q_{i,\ell}\, x_{i,\ell}.$$

In the approximation above, we interpret the coefficient $\eta^q_{i,\ell}$ 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 $\{\eta^q_{i,\ell} : i \in \mathcal{N},\ \ell \in \mathcal{T},\ q \in \mathcal{Q}\}$. For all $i \in \mathcal{N}$ and $t \in \mathcal{T}$, we set $\eta^{Q+1}_{i,\ell} = 0$. For each $q = Q, Q-1, \ldots, 1$, we execute the two steps.

- **<u>Construct Ideal Assortments</u>:** 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

$$A^q_{[s,f]} \;=\; \arg\max_{S\subseteq\mathcal{N}} \sum_{i\in\mathcal{N}} \phi^q_i(S)\Big[r_{i,[s,f]} \;-\; \sum_{h=s}^{f} \eta^{q+1}_{i,h}\Big]. \tag{6}$$

- **<u>Compute Opportunity Costs</u>:** 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

$$\eta^q_{i,\ell} = \sum_{[s,f]\in\mathcal{F}} \lambda^q_{[s,f]}\ \phi^q_i(A^q_{[s,f]})\, \frac{\mathbb{1}_{\{\ell\in[s,f]\}}}{f-s+1}\,\Big[r_{i,[s,f]} \;-\; \sum_{h=s}^{f}\eta^{q+1}_{i,h}\Big] \;+\; \eta^{q+1}_{i,\ell}. \tag{7}$$

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

In (6), $\sum_{\ell=s}^{f} \eta^{q+1}_{i,\ell}$ is the total opportunity cost of the capacities consumed by booking resource i for interval $[s,f]$ at time period q. Thus, $r_{i,[s,f]} - \sum_{\ell=s}^{f} \eta^{q+1}_{i,\ell}$ gives the revenue from booking resource i for interval $[s,f]$, net of the opportunity cost of the capacities consumed. In this case, $A^q_{[s,f]}$ 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 $\lambda^q_{[s,f]}$, we have a customer arriving at time period q with a booking request for interval $[s,f]$. If we offer the ideal assortment $A^q_{[s,f]}$, then this customer chooses resource i with probability $\phi^q_i(A^q_{[s,f]})$. At time period q, if we book resource i over interval $[s,f]$, then we obtain a net revenue of $r_{i,[s,f]} - \sum_{h=s}^{f} \eta^{q+1}_{i,h}$, but by doing so, we consume the capacity of resource i on each day $\ell \in \{s,\ldots,f\}$. Noting the fraction $\frac{\mathbb{1}_{\{\ell\in[s,f]\}}}{f-s+1}$, we spread the net revenue evenly over each day $\ell \in \{s,\ldots,f\}$.

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 $\{A^q_{[s,f]} : [s,f] \in \mathcal{F},\ q \in \mathcal{Q}\}$ computed through (6), we consider a static policy that always offers the assortment $A^q_{[s,f]}$ 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 $\mathsf{D}_{\max} = \max_{[s,f]\in\mathcal{F}} \{f - s + 1\}$, 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 $1/(2\,\mathsf{D}_{\max})$ fraction of the optimal total expected revenue.*

The proof of Theorem 4.1 is in Appendix B. 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 inherits the performance guarantee of $1/(2\,\mathsf{D}_{\max})$. Lastly, problem (6) has a combinatorial nature, but we can solve this problem efficiently under a variety of 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).

The coefficients $\{\eta^q_{i,\ell} : i \in \mathcal{N},\ \ell \in \mathcal{T},\ q \in \mathcal{Q}\}$ 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) and Baek and Ma (2021) use nonlinear value function approximations to give a policy with a performance guarantee of $1/(1 + \mathsf{D}_{\max})$, which is stronger than the performance guarantee of $1/(2\,\mathsf{D}_{\max})$ in Theorem 4.1. In our computational experiments, the performance of our linear value function approximations are comparable to that of nonlinear value function approximations. The competitive practical performance of our linear value function approximations, coupled with their interpretability, make them particularly appealing. In the next section, we use nonlinear value function approximations to give another static policy that improves the performance guarantee in Ma et al. (2020) and Baek and Ma (2021) even further.

5. 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 $\{\hat{J}^q_{\mathsf{P}} : q \in \mathcal{Q}\}$ of the form

$$\hat{J}^q_{\mathsf{P}}(\boldsymbol{x}) \;=\; \sum_{i\in\mathcal{N}} \sum_{[s,f]\in\mathcal{F}} \gamma^q_{i,[s,f]} \prod_{\ell=s}^{f} x_{i,\ell}.$$

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 $\{\gamma^q_{i,[s,f]} : i \in \mathcal{N},\ [s,f]\in\mathcal{F},\ q\in\mathcal{Q}\}$, 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 an *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 $[s,f]\cap[a,b]\neq\varnothing$, then we have $C_{[s,f]}\cap[a,b]\neq\varnothing$. 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 an intersection preserving subset $C_{[s,f]}$ is to set $C_{[s,f]} = \{s,\ldots,f\}$. Using intersection preserving subsets with fewer elements will yield better performance guarantees. Later in this section, we discuss finding the smallest intersection preserving subsets and give nontrivial examples of 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 $\{\gamma^q_{i,[s,f]} : i \in \mathcal{N},\ [s,f]\in\mathcal{F},\ q\in\mathcal{Q}\}$ as follows. For all $i\in\mathcal{N}$ and $[s,f]\in\mathcal{F}$, set $\gamma^{Q+1}_{i,[s,f]} = 0$. For each $q = Q, Q-1,\ldots,1$, 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

$$B^q_{[s,f]} \;=\; \arg\max_{S\subseteq\mathcal{N}} \left\{ \sum_{i\in\mathcal{N}} \phi^q_i(S) \left[r_{i,[s,f]} \;-\; \sum_{[a,b]\in\mathcal{F}} \Big| [s,f]\cap C_{[a,b]} \Big| \; \gamma^{q+1}_{i,[a,b]} \right] \right\}. \tag{8}$$

- **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

$$\gamma^q_{i,[s,f]} \;=\; \lambda^q_{[s,f]}\;\phi^q_i(B^q_{[s,f]}) \left[r_{i,[s,f]} \;-\; \sum_{[a,b]\in\mathcal{F}} \Big| [s,f]\cap C_{[a,b]} \Big| \; \gamma^{q+1}_{i,[a,b]} \right] \;+\; \gamma^{q+1}_{i,[s,f]}. \tag{9}$$

In (8)-(9), $|C|$ is the cardinality of set $C\subseteq\mathcal{T}$. The algorithm above fully specifies the coefficients $\{\gamma^q_{i,[s,f]} : i \in \mathcal{N},\ [s,f]\in\mathcal{F},\ q\in\mathcal{Q}\}$. We give some intuition for the algorithm. Given that all resources are available on all days at time period q, $J^q(\boldsymbol{e})$ is the optimal total expected revenue from booking requests over time periods $\{q,\ldots,Q\}$. We approximate this optimal total expected revenue by $\hat{J}^q_{\mathsf{P}}(\boldsymbol{e}) = \sum_{i\in\mathcal{N}}\sum_{[s,f]\in\mathcal{F}} \gamma^q_{i,[s,f]}$. Thus, we intuitively interpret $\gamma^q_{i,[s,f]}$ as an approximation to the optimal total expected revenue over time periods $\{q,\dots,Q\}$ from booking requests for resource i over interval $[s,f]$. In (8), by the definition of an intersection preserving subset, if $[s,f]\cap[a,b]\neq\varnothing$, then $[s,f]\cap C_{[a,b]}\neq\varnothing$ as well. Also, if $[s,f]\cap[a,b]\neq\varnothing$, 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 $[a,b]$. So, we can use $\sum_{[a,b]\in\mathcal{F}} \mathbb{1}_{\{[s,f]\cap C_{[a,b]}\neq\varnothing\}}\,\gamma^{q+1}_{i,[a,b]}$ to capture the opportunity cost of the booking requests that we can no longer accept by booking resource i for interval $[s,f]$. We use $\sum_{[a,b]\in\mathcal{F}}\Big|[s,f]\cap C_{[a,b]}\Big|\,\gamma^{q+1}_{i,[a,b]}$ as an upper bound approximation to this opportunity cost. In this case, $B^q_{[s,f]}$ is the assortment that maximizes the net expected revenue from a customer making a booking request for interval $[s,f]$ at time period q. Using the upper bound $\sum_{[a,b]\in\mathcal{F}}\Big|[s,f]\cap C_{[a,b]}\Big|\,\gamma^{q+1}_{i,[a,b]}$ on the opportunity cost allows us to upper bound the optimal total expected revenue in Proposition C.3. In (9), we accumulate our approximation to the optimal total expected revenue over time periods $\{q,\dots,Q\}$ from booking requests for resource i over interval $[s,f]$. On the right side of (9), the first term captures the net expected revenue at time period q under the ideal assortment $B^q_{[s,f]}$ and 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 $\{B^q_{[s,f]}:[s,f]\in\mathcal{F},\ q\in\mathcal{Q}\}$ computed through (8), we consider a static policy that always offers the assortment $B^q_{[s,f]}$ 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 5.1 (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.*

We give the proof of Theorem 5.1 in Appendix C. Similar to Theorem 4.1, the proof of Theorem 5.1 uses the value function approximations $\{\hat{J}^q_{\mathsf{P}}:q\in\mathcal{Q}\}$ to upper bound the performance of the optimal policy and to lower bound the performance of the itinerary based static policy, but the specifics of the proof uses properties of intersection preserving subsets. In Appendix D, we show that the performance guarantee in Theorem 5.1 is tight. Recall that setting $C_{[s,f]} = \{s, \ldots, f\}$ trivially yields an intersection preserving subset for the interval $[s, f]$. For this intersection preserving subset, we have $|C_{[s,f]}| = f - s + 1 \leq \mathsf{D}_{\max}$, so there exists a collection of intersection preserving subsets whose norm is at most $\mathsf{D}_{\max}$. By Theorem 5.1, the itinerary based static policy that we obtain by using such a trivial collection of intersection preserving subsets has a performance guarantee of $1/(1 + \mathsf{D}_{\max})$. In the next theorem, letting $\mathsf{D}_{\min} = \min_{[s,f]\in\mathcal{F}}\{f - s + 1\}$ to capture the minimum use duration for any possible booking request, we show that there exists a collection of intersection preserving subsets whose norm is at most $1 + \lceil(\mathsf{D}_{\max} - 1)/\mathsf{D}_{\min}\rceil$ and 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 $\|\mathcal{C}\| = \max_{[s,f]\in\mathcal{F}} |C_{[s,f]}|$, 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

$$\begin{aligned} \min \Big\{ t \ : \ t \ &\geq \ \Big|C_{[s,f]}\Big| \quad \forall\, [s,f] \in \mathcal{F}, \\ &\Big|C_{[s,f]} \cap [a,b]\Big| \ \geq \ 1 \quad \forall\, [s,f] \text{ and } [a,b] \in \mathcal{F} \text{ such that } [s,f] \cap [a,b] \neq \varnothing, \\ &C_{[s,f]} \subseteq [s,f] \quad \forall\, [s,f] \in \mathcal{F}, \quad t \geq 0 \Big\}. \end{aligned} \tag{10}$$

By the first constraint, we have $t = \max_{[s,f]\in\mathcal{F}} |C_{[s,f]}|$ in an optimal solution to problem (10). The second and third constraints ensure that $C_{[s,f]}$ is an intersection preserving subset.

Theorem 5.2 (Smallest Norm Collection) *The optimal objective value of problem* (10) *is at most* $1 + \lceil(\mathsf{D}_{\max} - 1)/\mathsf{D}_{\min}\rceil$. *Furthermore, we can obtain an optimal solution to this problem by solving a linear program with* $O(\mathsf{D}_{\max}^2 T)$ *decision variables and* $O(\mathsf{D}_{\max}^3 T)$ *constraints.*

We give the proof of Theorem 5.2 in Appendix E. By Theorem 5.2, there always exists a collection of intersection preserving subsets whose norm is at most $1 + \lceil(\mathsf{D}_{\max} - 1)/\mathsf{D}_{\min}\rceil$. By Theorem 5.1, the corresponding itinerary based static policy has a performance guarantee of $\frac{1}{2+\lceil(\mathsf{D}_{\max}-1)/\mathsf{D}_{\min}\rceil}$. Considering a more general case with non-unit resource capacities, Baek and Ma (2021) give a policy with a performance guarantee of $\frac{1}{1+\mathsf{D}_{\max}}$. If $\mathsf{D}_{\max} > \mathsf{D}_{\min} > 1$, then $\frac{1}{2+\lceil(\mathsf{D}_{\max}-1)/\mathsf{D}_{\min}\rceil} > \frac{1}{1+\mathsf{D}_{\max}}$, so we get a stronger performance guarantee by focusing on resources with unit capacities. There are two other significant differences for the two papers. First, Baek and Ma (2021) use a recursion similar to the one in (9) to capture the opportunity cost of a resource, but they obtain the analogue of the assortment $B^q_{[s,f]}$ by solving a linear programming approximation; see Section 4.1 in Baek and Ma (2021). This approximation works well in an asymptotic regime where the resource capacities are large, but leveraging the linear program when the resources have unit capacities yields a policy that does not perform as well in practice. Second, their policy is not static, so we cannot perform rollout on their policy, which is critical for boosting practical performance. Closing this section, we give examples of intersection preserving subsets in two special settings.

First, consider the setting where there exists a set of days $\{\tau_1, \ldots, \tau_K\}$ such that each booking request starts on one of these days. In this case, we can show that $C_{[s,f]} = [s,f] \cap \{\tau_1, \ldots, \tau_K\}$ is an intersection preserving subset for the interval $[s,f]$. Second, consider the setting where each booking request is for the same number of days, so $s - f = L$ for all $[s,f] \in \mathcal{F}$. In this case, we can show that $C_{[s,f]} = \{s,f\}$ is an intersection preserving subset for the interval $[s,f]$. In Appendix F, we show that these two examples are indeed intersection preserving subsets in their respective settings. By Theorem 5.2, we always have a collection of intersection preserving subsets with norm at most $1 + \lceil (\mathsf{D}_{\max} - 1)/\mathsf{D}_{\min} \rceil$, but the norms in these special settings may be smaller.

6. 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.

6.1 Resource Based Problems Through Revenue Allocations

The starting point for our upper bound on the optimal total expected revenue is a linear programming approximation that is formulated under the assumption that the arrivals and choices of the customers take on their expected values. To formulate such an approximation, we use two sets of decision variables. First, we use the decision variable $h^q_{[s,f]}(S)$ 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 $y^q_{i,[s,f]}$ 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

$$
\begin{aligned}
\mathsf{Z}^*_{\mathsf{LP}} \;=\; \max \quad & \sum_{q \in \mathcal{Q}} \sum_{i \in \mathcal{N}} \sum_{[s,f] \in \mathcal{F}} r_{i,[s,f]} \, y^q_{i,[s,f]} \\
\text{st} \quad & \sum_{q \in \mathcal{Q}} \sum_{[s,f] \in \mathcal{F}} \sum_{S \subseteq \mathcal{N}} \mathbb{1}_{\{\ell \in [s,f]\}} \, \phi^q_i(S) \, h^q_{[s,f]}(S) \;\leq\; 1 \qquad \forall\, i \in \mathcal{N},\ \ell \in \mathcal{T} \\
& \sum_{S \subseteq \mathcal{N}} h^q_{[s,f]}(S) \;=\; \lambda^q_{[s,f]} \qquad \forall\, [s,f] \in \mathcal{F},\ q \in \mathcal{Q} \\
& \sum_{S \subseteq \mathcal{N}} \phi^q_i(S) \, h^q_{[s,f]}(S) \;=\; y^q_{i,[s,f]} \qquad \forall\, i \in \mathcal{N},\ [s,f] \in \mathcal{F},\ q \in \mathcal{Q} \\
& h^q_{[s,f]}(S) \;\geq\; 0 \qquad \forall\, [s,f] \in \mathcal{F},\ S \subseteq \mathcal{N},\ q \in \mathcal{Q}.
\end{aligned}
\tag{11}
$$

We do not explicitly impose nonnegativity on the decision variable $y^q_{i,[s,f]}$ because the third and fourth constraints above ensure this constraint. In the objective function, we accumulate the total expected revenue from the bookings. By the first constraint, the expected capacity consumption of resource i on day ℓ does not exceed one. By the second constraint, the 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 programs as in (11) are commonly used to obtain upper bounds on optimal expected revenues. Such upper bounds are asymptotically tight as the resource capacities and the expected customer arrivals increase at the same rate. In our problem, the capacities of all resources are invariably one, making such an asymptotic regime irrelevant. In our computational experiments, the upper bound from the linear program in (11) can indeed be quite loose. Here, we give a different upper bound. 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 different resources.

Let $\beta^q_{i,[s,f]\to j}$ 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 $\sum_{j\in\mathcal{N}} \beta^q_{i,[s,f]\to j} = r_{i,[s,f]}$. In our approach, we solve a separate revenue management problem for each resource and add 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 $\beta^q_{i,[s,f]\to j}$. We capture the state of resource j by using the vector $\boldsymbol{x}_j = (x_{j,1},\ldots,x_{j,T}) \in \{0,1\}^T$, where $x_{j,\ell} = 1$ if and only if resource j is available for use on day ℓ. Noting that $\boldsymbol{e}_{[s,f]} \in \{0,1\}^T$ is the vector with ones in the components corresponding to days $\{s,\ldots,f\}$, we can find the optimal policy in the revenue management problem for resource j by computing the value functions $\{V^q_{\boldsymbol{\beta},j} : q \in \mathcal{Q}\}$ through the dynamic program

$$V^q_{\boldsymbol{\beta},j}(\boldsymbol{x}_j) = \sum_{[s,f]\in\mathcal{F}} \lambda^q_{[s,f]} \max_{S\subseteq\mathcal{N}} \Bigg\{ \phi^q_j(S) \left(\prod_{\ell=s}^{f} x_{j,\ell} \right) \Big[\beta^q_{j,[s,f]\to j} + V^{q+1}_{\boldsymbol{\beta},j}(\boldsymbol{x}_j - \boldsymbol{e}_{[s,f]}) - V^{q+1}_{\boldsymbol{\beta},j}(\boldsymbol{x}_j) \Big] + \sum_{i\in\mathcal{N}\setminus\{j\}} \phi^q_i(S)\, \beta^q_{i,[s,f]\to j} \Bigg\} + V^{q+1}_{\boldsymbol{\beta},j}(\boldsymbol{x}_j), \quad (12)$$

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

In (12), at time period q, if we offer the assortment S of resources to a booking request for interval $[s,f]$, then the customer chooses resource j with probability $\phi^q_j(S)$. If we have capacity for resource j

to accommodate the booking request, then we make the revenue of $\beta^q_{j,[s,f]\to j}$. The opportunity cost of the consumed capacities is $V^{q+1}_{\boldsymbol{\beta},j}(\boldsymbol{x}_j) - V^{q+1}_{\boldsymbol{\beta},j}(\boldsymbol{x}_j - \boldsymbol{e}_{[s,f]})$. We have infinite capacity for all resources other than resource j, so if the customer chooses some other resource i, then we make the revenue of $\beta^q_{i,[s,f]\to j}$, but the opportunity cost of the consumed capacities is zero. Because booking resource i brings the revenue of $\beta^q_{i,[s,f]\to j}$, it may indeed be optimal to offer resource $i \in \mathcal{N} \setminus \{j\}$. In the next proposition, we show that we get an upper bound on the value functions $\{J^q : q \in \mathcal{Q}\}$ in (1) by solving the dynamic program in (12). We defer all proofs in this section to Appendix G.

Proposition 6.1 (Decomposition Upper Bound) *For any revenue allocations $\boldsymbol{\beta}$ that satisfy $\sum_{j\in\mathcal{N}} \beta^q_{i,[s,f]\to j} = r_{i,[s,f]}$ for all $i \in \mathcal{N}$, $[s,f] \in \mathcal{F}$ and $q \in \mathcal{Q}$, letting the value functions $\{V^q_{\boldsymbol{\beta},j} : q \in \mathcal{Q}\}$ be computed through* (12)*, for each $\boldsymbol{x} = (\boldsymbol{x}_j : j \in \mathcal{N}) \in \{0,1\}^{N\times T}$ and $q \in \mathcal{Q}$, we have*

$$\sum_{j\in\mathcal{N}} V^q_{\boldsymbol{\beta},j}(\boldsymbol{x}_j) \;\geq\; J^q(\boldsymbol{x}).$$

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

6.2 Choosing the Revenue Allocations

In problem (11), duplicating the decision variable $h^q_{[s,f]}$ for each resource j to obtain the decision variables $\{h^q_{[s,f]\to j} : j \in \mathcal{N}\}$, we consider a variant of this problem given by

$$\begin{aligned}
\max \quad & \sum_{q\in\mathcal{Q}} \sum_{i\in\mathcal{N}} \sum_{[s,f]\in\mathcal{F}} r_{i,[s,f]}\, y^q_{i,[s,f]} && (13)\\
\text{st} \quad & \sum_{q\in\mathcal{Q}} \sum_{[s,f]\in\mathcal{F}} \sum_{S\subseteq\mathcal{N}} \mathbb{1}_{\{\ell\in[s,f]\}}\, \phi^q_i(S)\, h^q_{[s,f]\to i}(S) \;\leq\; 1 \qquad \forall\, i\in\mathcal{N},\ \ell\in\mathcal{T} \\
& \sum_{S\subseteq\mathcal{N}} h^q_{[s,f]\to j}(S) \;=\; \lambda^q_{[s,f]} \qquad \forall\, j\in\mathcal{N},\, [s,f]\in\mathcal{F},\ q\in\mathcal{Q} \\
& \sum_{S\subseteq\mathcal{N}} \phi^q_i(S)\, h^q_{[s,f]\to j}(S) \;=\; y^q_{i,[s,f]} \qquad \forall\, i,j\in\mathcal{N},\ [s,f]\in\mathcal{F},\ q\in\mathcal{Q} \\
& h^q_{[s,f]\to j}(S) \;\geq\; 0 \qquad \forall\, j\in\mathcal{N},\ [s,f]\in\mathcal{F},\ S\subseteq\mathcal{N},\ q\in\mathcal{Q}.
\end{aligned}$$

In Lemma G.1 in Appendix G, we show that the problem above has the same optimal objective value as the linear programming approximation in (11). Intuitively speaking, although problem (13) duplicates the decision variable $h^q_{[s,f]}$ in problem (11) for each resource j, it does not duplicate the decision variable $y^q_{i,[s,f]}$. Because the objective functions of both of these two problems depend only on the decision variables $\{y^q_{i,[s,f]} : i \in \mathcal{N},\ [s,f] \in \mathcal{F},\ q \in \mathcal{Q}\}$, the two problems end up having the same optimal objective value. We use the dual solution to problem (13) to choose the revenue allocations. Associating the dual variables $\{\beta^q_{i,[s,f]\to j} : i,j \in \mathcal{N},\ [s,f] \in \mathcal{F},\ q \in \mathcal{Q}\}$ with the third constraint in problem (13), in the dual of problem (13), the constraint associated with the decision variable $y^q_{i,[s,f]}$ is $\sum_{j\in\mathcal{N}} \beta^q_{i,[s,f]\to j} = r_{i,[s,f]}$. Thus, letting $\hat{\boldsymbol{\beta}} = \{\hat{\beta}^q_{i,[s,f]\to j} : i,j\in\mathcal{N},\ [s,f]\in\mathcal{F},\ q\in\mathcal{Q}\}$ be the optimal values of the dual variables associated with the third constraint in problem (13), 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 (12), 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 (11). In this theorem, we use $\boldsymbol{e}' \in \{0,1\}^T$ to denote the vector of all ones.

Theorem 6.2 (Choice of Revenue Allocations) *Letting $\hat{\boldsymbol{\beta}}$ be the optimal values of the dual variables associated with the third constraint in problem* (13) *and the value functions $\{V^q_{\hat{\boldsymbol{\beta}},j} : q\in\mathcal{Q}\}$ be computed through* (12) *with the revenue allocations $\hat{\boldsymbol{\beta}}$, we have*

$$\sum_{j\in\mathcal{N}} V^1_{\hat{\boldsymbol{\beta}},j}(\boldsymbol{e}') \ \leq\ \mathsf{Z}^*_{\mathsf{LP}}.$$

Noting Proposition 6.1, we get $\mathsf{Z}^*_{\mathsf{LP}} \geq \sum_{j\in\mathcal{N}} V^1_{\hat{\boldsymbol{\beta}},j}(\boldsymbol{e}') \geq J^1(\boldsymbol{e})$, so $\sum_{j\in\mathcal{N}} V^1_{\hat{\boldsymbol{\beta}},j}(\boldsymbol{e}')$ 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 (11).

6.3 Computing the Upper Bound

In (12), 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 $\boldsymbol{x}_j = (x_{j,1},\ldots,x_{j,T}) \in \{0,1\}^T$ 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 $\boldsymbol{x}_j$ 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, $\boldsymbol{x}_{j,\ell} = 0$ for all $j\in\{a,\ldots,b\}$, $x_{j,a-1} = 1$, 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 $\Gamma^q_{\boldsymbol{\beta},j}(a,b)$ be the total expected revenue obtained over time periods $\{q,\ldots,Q\}$ from the booking requests for intervals that start in the interval $[a,b]$. We can compute the value functions $\{\Gamma^q_{\boldsymbol{\beta},j} : q\in\mathcal{Q}\}$ as

$$\Gamma^q_{\boldsymbol{\beta},j}(a,b) \ =\ \sum_{k=q}^{Q} \sum_{[s,f]\in\mathcal{F}} \mathbb{1}_{\{s\in[a,b]\}}\, \lambda^k_{[s,f]} \max_{S\subseteq\mathcal{N}\setminus\{j\}} \left\{ \sum_{i\in\mathcal{N}\setminus\{j\}} \phi^k_i(S)\ \beta^k_{i,[s,f]\to j} \right\}. \tag{14}$$

To interpret (14), 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 $\{q,\ldots,Q\}$. On the other hand, recall that we refer to the interval $[a,b]$ as a maximal available interval with respect to $\boldsymbol{x}_j$ 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 $\Theta^q_{\boldsymbol{\beta},j}(a,b)$ be the total expected revenue obtained over time periods $\{q,\ldots,Q\}$ from the booking requests for intervals that start in the interval $[a,b]$. We can compute the value functions $\{\Theta^q_{\boldsymbol{\beta},j} : q\in\mathcal{Q}\}$ as

$$\begin{aligned}\Theta^q_{\boldsymbol{\beta},j}(a,b) \;=\; &\sum_{[s,f]\in\mathcal{F}} \lambda^q_{[s,f]}\, \mathbb{1}_{\{s\in[a,b],\,[s,f]\subseteq[a,b]\}} \max_{S\subseteq\mathcal{N}} \Bigg\{ \sum_{i\in\mathcal{N}\setminus\{j\}} \phi^q_i(S)\,\beta^q_{i,[s,f]\to j} \\ &\qquad + \phi^q_j(S)\Big[\beta^q_{j,[s,f]\to j} + \Theta^{q+1}_{\boldsymbol{\beta},j}(a,s-1) + \Theta^{q+1}_{\boldsymbol{\beta},j}(f+1,b) + \Gamma^{q+1}_{\boldsymbol{\beta},j}(s,f) - \Theta^{q+1}_{\boldsymbol{\beta},j}(a,b)\Big]\Bigg\} \\ &+ \sum_{[s,f]\in\mathcal{F}} \lambda^q_{[s,f]}\, \mathbb{1}_{\{s\in[a,b],\,[s,f]\not\subseteq[a,b]\}} \max_{S\subseteq\mathcal{N}\setminus\{j\}} \Bigg\{ \sum_{i\in\mathcal{N}\setminus\{j\}} \phi^q_i(S)\,\beta^q_{i,[s,f]\to j}\Bigg\} + \Theta^{q+1}_{\boldsymbol{\beta},j}(a,b). \end{aligned} \tag{15}$$

To interpret (15), 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 $\{s,\ldots,f\}$, 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 $\mathcal{I}(\boldsymbol{x}_j)$ is the collection of maximal available intervals with respect to $\boldsymbol{x}_j$. Let $\mathcal{H}(\boldsymbol{x}_j)$ be the collection of maximal unavailable intervals with respect to $\boldsymbol{x}_j$. In the top portion of Figure 1, for example, for the value of $\boldsymbol{x}_j$ in this figure, we have $\mathcal{H}(\boldsymbol{x}_j)=\{[1,1],[5,5],[11,12]\}$. In the next theorem, we show that we can solve the dynamic program in (15) to get a solution for (12).

Theorem 6.3 (Intervals) *Letting the value function* $\{V^q_{\boldsymbol{\beta},j} : q\in\mathcal{Q}\}$ *be computed through* (12)*, for each* $\boldsymbol{x}_j\in\{0,1\}^T$ *and* $q\in\mathcal{Q}$*, we have* $V^q_{\boldsymbol{\beta},j}(\boldsymbol{x}_j)=\sum_{[a,b]\in\mathcal{I}(\boldsymbol{x}_j)}\Theta^q_{\boldsymbol{\beta},j}(a,b)+\sum_{[a,b]\in\mathcal{H}(\boldsymbol{x}_j)}\Gamma^q_{\boldsymbol{\beta},j}(a,b)$.

In Appendix H, we also argue that we can obtain the dynamic program in (12) by using Lagrangian relaxation on (1), connecting our revenue allocations to Lagrange multipliers.

7. 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.

7.1 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 $\{s,\ldots,f\}$.

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 $\mathcal{T}=\{1,\ldots,70\}$, so we have 10 weeks in the booking horizon. Customers arrive over time periods $\mathcal{Q}=\{1,\ldots,700\}$. The set of possible intervals of stay is $\mathcal{F}=\{[a,b]: a\leq b\leq a+\mathsf{D}_{\max}-1\}$, where $\mathsf{D}_{\max}$ 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 $\{\lambda^q_{[s,f]}:[s,f]\in\mathcal{F},\ q\in\mathcal{Q}\}$, for each $[s,f]\in\mathcal{F}$ and $q\in\mathcal{Q}$, we sample a weight $\beta^q_{[s,f]}$ from the uniform distribution over $[0,\overline{U}^q_{[s,f]}]$ and normalize the weights by setting $\gamma^q_{[s,f]}=\frac{\beta^q_{[s,f]}}{\sum_{[a,b]\in\mathcal{F}}\beta^q_{[a,b]}}$. In this case, using $\{\gamma^q_{[s,f]}:[s,f]\in\mathcal{F},\ q\in\mathcal{Q}\}$ 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 $\mathsf{Demand}=\sum_{i\in\mathcal{N}}\sum_{q\in\mathcal{Q}}\sum_{[s,f]\in\mathcal{F}}\gamma^q_{[s,f]}\,\phi^q_i(\mathcal{N})\,(f-s+1)$. 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 $\lambda^q_{[s,f]}=\rho\,\gamma^q_{[s,f]}\,\frac{NT}{\mathsf{Demand}}$, 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 $\overline{U}^q_{[s,f]}=\mathsf{D}_{\max}-(f-s)$ for $q=1,\ldots,233$, $\overline{U}^q_{[s,f]}=1$ for $q=234,\ldots,466$ and $\overline{U}^q_{[s,f]}=f-s+1$ for $q=467,\ldots,700$. 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 $\phi^q_i(S)=\frac{v_i}{v_0+\sum_{j\in S}v_j}$. 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 $\mathsf{D}_{\max} \in \{6,8,10\}$, $\rho \in \{1.0,1.2,1.4\}$, and $\delta \in \{0.7,0.9\}$, we obtain 18 parameter configurations for our test problems.

Benchmark Policies: Our benchmarks 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 benchmarks 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 $1/(2\,\mathsf{D}_{\max})$, but being 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 $J^{q+1}(\boldsymbol{x})$ on the right side of (1) with $\hat{J}^{q+1}_{\mathsf{L}}(\boldsymbol{x}) = \sum_{i\in\mathcal{N}}\sum_{\ell\in\mathcal{T}} \eta^{q+1}_{i,\ell}\, x_{i,\ell}$, if the state of the system at time period q is $\boldsymbol{x}$ 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 $\hat{S}^q_{\mathsf{L}}(\boldsymbol{x}) = \arg\max_{S\subseteq\mathcal{N}} \sum_{i\in\mathcal{N}} \phi^q_i(S)\left(\prod_{\ell=s}^{f} x_{i,\ell}\right)\left(r_{i,[s,f]} - \sum_{\ell=s}^{f}\eta^{q+1}_{i,\ell}\right)$. 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 $1/(2\,\mathsf{D}_{\max})$. The key is to observe that an analogue of the chain of inequalities in the proof of Proposition B.5 still holds when we replace $A^q_{[s,f]}$ with $\hat{S}^q_{\mathsf{L}}(\boldsymbol{x})$.

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 $\{\eta^q_{i,\ell} : i\in\mathcal{N},\ \ell\in\mathcal{T},\ q\in\mathcal{Q}\}$ 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 $\boldsymbol{x}$, then we set $\eta^q_{i,\ell} = 0$ 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 benchmark as LIN5. Lastly, we use the rollout policy from the resource based static policy. We refer to this benchmark as LINR. By the discussion at the end of Section 3, this policy inherits the performance guarantee of $1/(2\,\mathsf{D}_{\max})$ 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 $\{J^q_{\mathsf{L}} : q\in\mathcal{Q}\}$, whereas noting that $\{J^q_{\boldsymbol{\mu}} : q\in\mathcal{Q}\}$ in (5) is not separable, LINR does not necessarily use a separable approximation.

Polynomial Approximations (POL1, POL5, POLR). The benchmarks POL1, POL5, and POLR are the analogues of LIN1, LIN5, and LINR, but they use the polynomial value function approximations presented in Section 5. As in LIN1, POL1 is the greedy policy with respect to the polynomial approximations. We can show that it shares the guarantee of $\frac{1}{2+\lceil(\mathsf{D}_{\max}-1)/\mathsf{D}_{\min}\rceil}$ with the itinerary based static policy. In POL5, when recomputing the coefficients $\{\gamma^q_{i,[s,f]} : i \in \mathcal{N},\ [s,f] \in \mathcal{F},\ q \in \mathcal{Q}\}$ at the beginning of a segment, if the state of the system is $\boldsymbol{x}$, then we set $\gamma^q_{i,[s,f]} = 0$ 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 5. Lastly, POLR is the rollout of the itinerary based static policy.

Linear Programming Approximation (LP1, LP5, LPR). We use three benchmarks based on the linear program in (11). The first constraint in (11) ensures that the total expected capacity consumption of each resource on each day does not exceed one. Letting $\{\hat{\mu}_{i,\ell} : i \in \mathcal{N},\ \ell \in \mathcal{T}\}$ be the optimal values of the dual variables for the first constraint, we use $\hat{\mu}_{i,\ell}$ as the opportunity cost of the capacity of resource i on day ℓ, yielding the value function approximations $\{\hat{J}^q_{\mathsf{LP}} : q \in \mathcal{Q}\}$ with $\hat{J}^q_{\mathsf{LP}}(\boldsymbol{x}) = \sum_{i\in\mathcal{N}} \sum_{\ell\in\mathcal{T}} \hat{\mu}_{i,\ell}\, x_{i,\ell}$. In this case, LP1 follows the greedy policy with respect to the value function approximations $\{\hat{J}^q_{\mathsf{LP}} : q \in \mathcal{Q}\}$. Similar to LIN5 and POL5, the benchmark LP5 splits the selling horizon into five equal segments and recomputes the opportunity costs at the beginning of each segment by solving problem (11) with the current values of remaining capacities. In LPR, we extract a static policy from problem (11) and use the rollout policy from this static policy. In particular, letting $\{\hat{h}^q_{[s,f]}(S) : [s,f] \in \mathcal{F},\ S \subseteq \mathcal{N},\ q \in \mathcal{Q}\}$ and $\{\hat{y}^q_{i,[s,f]} : i \in \mathcal{N},\ [s,f] \in \mathcal{F},\ q \in \mathcal{Q}\}$ be an optimal solution to problem (11), if we have a booking request for interval $[s,f]$ at time period q, then the static policy offers the assortment S of resources with probability $\hat{h}^q_{[s,f]}(S)/\lambda^q_{[s,f]}$. In this case, LPR is the rollout policy corresponding to this static policy.

Greedy Policy from the Upper Bound (GUB). This benchmark builds on the dynamic program in (12). We choose the revenue allocations $\hat{\boldsymbol{\beta}} = \{\hat{\beta}^q_{i,[s,f]\to j} : i,j \in \mathcal{N},\ [s,f] \in \mathcal{F},\ q \in \mathcal{Q}\}$ by using problem (13) as discussed in Section 6.2. Using these revenue allocations in the dynamic program in (12), we compute the value functions $\{V^q_{\hat{\boldsymbol{\beta}},j} : q \in \mathcal{Q}\}$ for each $j \in \mathcal{N}$, yielding the value function approximations $\{\hat{J}^q_{\mathsf{GUB}} : q \in \mathcal{Q}\}$ with $\hat{J}^q_{\mathsf{GUB}}(\boldsymbol{x}) = \sum_{j\in\mathcal{N}} V^q_{\hat{\boldsymbol{\beta}},j}(\boldsymbol{x}_j)$. In this case, GUB follows the greedy policy with respect to the value function approximations $\{\hat{J}^q_{\mathsf{GUB}} : q \in \mathcal{Q}\}$.

Dynamic Programming Decomposition (DEC). This benchmark is the standard approach to heuristically decompose the dynamic program in (1) by each resource and day combination; see, for example, Liu and van Ryzin (2008). Note that each resource has one unit of capacity on each day. Thus, DEC constructs linear value function approximations $\{\hat{J}^q_{\mathsf{Dec}} : q \in \mathcal{Q}\}$ of the form $\hat{J}^q_{\mathsf{Dec}}(\boldsymbol{x}) = \sum_{i\in\mathcal{N}} \sum_{\ell\in\mathcal{T}} \zeta^q_{i,\ell}\, x_{i,\ell}$, for some opportunity costs $\{\zeta^q_{i,\ell} : i \in \mathcal{N},\ \ell \in \mathcal{T},\ q \in \mathcal{Q}\}$. In this case, we follow the greedy policy with respect to these approximations.

Comparison of Upper Bounds: The optimal objective value $\mathsf{Z}^*_{\mathsf{LP}}$ of the linear programming approximation in (11) provides an upper bound on the optimal total expected revenue. By

Params. $(\mathsf{D}_{\max}, \rho, \delta)$	% Gap in Bounds
$(6, 1.0, 0.9)$	19.59
$(6, 1.0, 0.7)$	27.71
$(6, 1.2, 0.9)$	15.38
$(6, 1.2, 0.7)$	11.99
$(6, 1.4, 0.9)$	18.39
$(6, 1.4, 0.7)$	24.66
Avg.	19.62

Params. $(\mathsf{D}_{\max}, \rho, \delta)$	% Gap in Bounds
$(8, 1.0, 0.9)$	27.16
$(8, 1.0, 0.7)$	25.63
$(8, 1.2, 0.9)$	25.62
$(8, 1.2, 0.7)$	15.53
$(8, 1.4, 0.9)$	24.78
$(8, 1.4, 0.7)$	24.52
Avg.	23.87

Params. $(\mathsf{D}_{\max}, \rho, \delta)$	% Gap in Bounds
$(10, 1.0, 0.9)$	9.28
$(10, 1.0, 0.7)$	20.14
$(10, 1.2, 0.9)$	13.82
$(10, 1.2, 0.7)$	12.93
$(10, 1.4, 0.9)$	28.17
$(10, 1.4, 0.7)$	29.48
Avg.	18.97

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

Proposition 6.1, we can solve the dynamic program in (12) to also obtain an upper bound on the optimal total expected revenue. Furthermore, by Theorem 6.2, we can use the optimal dual solution to problem (13) to choose the revenue allocations such that the upper bound we obtain is at least as tight as that provided by $\mathsf{Z}^*_{\mathsf{LP}}$. In particular, letting $\hat{\boldsymbol{\beta}}$ be the optimal values of the dual variables associated with the third constraint in problem (13), $\sum_{j\in\mathcal{N}} V^1_{\hat{\boldsymbol{\beta}},j}(\boldsymbol{e}')$ is an upper bound on the optimal total expected revenue, and this upper bound is at least as tight as $\mathsf{Z}^*_{\mathsf{LP}}$. In Table 1, we compare the upper bounds $\mathsf{Z}^*_{\mathsf{LP}}$ and $\sum_{j\in\mathcal{N}} V^1_{\hat{\boldsymbol{\beta}},j}(\boldsymbol{e}')$. In the table, the first column shows the parameter configuration for each test problem. The second column shows the percent gap $100 \times \frac{\mathsf{Z}^*_{\mathsf{LP}} - \sum_{j\in\mathcal{N}} V^1_{\hat{\boldsymbol{\beta}},j}(\boldsymbol{e}')}{\sum_{j\in\mathcal{N}} V^1_{\hat{\boldsymbol{\beta}},j}(\boldsymbol{e}')}$ 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 29.48%. In our approach, once we allocate the revenue associated with each itinerary over different resources, the dynamic program that we solve for each resource incorporates the uncertainty in the customer arrivals and choices, giving our approach its edge.

Another approach to obtain an upper bound in a general revenue management problem is based on decomposing the dynamic program for the problem by the components of the state vector; see Topaloglu (2009). Noting that our state vector is $\boldsymbol{x} = (x_{i,\ell} : i \in \mathcal{N},\ \ell \in \mathcal{T})$, this general upper bound uses value function approximations of the form $\hat{J}^q_{\mathsf{Gen}}(\boldsymbol{x}) = \sum_{i\in\mathcal{N}} \sum_{\ell\in\mathcal{T}} \nu^q_{i,\ell}(x_{i,\ell})$. In contrast, our upper bound uses value function approximations of the form $\hat{J}^q_{\mathsf{GUB}}(\boldsymbol{x}) = \sum_{j\in\mathcal{N}} V^q_{\boldsymbol{\beta},j}(\boldsymbol{x}_j)$. Thus, the general upper bound assumes that the value functions are separable by both resources and days of use, whereas our upper bound assumes that the value functions are separable only by resources. The optimal value functions are not necessarily separable by resources or days of use, but the general upper bound takes the separability assumption one step further, losing the interactions between different days of use. We computed the general upper bound for our test problems and it was almost as poor as the upper bound from the linear programming approximation. While the general upper bound applies to any revenue management problem, we exploit the unit capacity of the resources and the interval structure of the requests to obtain substantially tighter bounds.

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.	Total Exp. Revenue										
$(\mathsf{D}_{\max}, \rho, \delta)$	LIN1	POL1	LP1	LIN5	POL5	LP5	LINR	POLR	LPR	GUB	DEC
$(6, 1.0, 0.9)$	74.46	71.21	73.40	77.02	72.56	75.37	78.00	78.17	**78.55**	78.40	74.04
$(6, 1.0, 0.7)$	80.66	75.34	78.17	82.33	76.89	80.48	83.41	83.54	83.87	**84.94**	80.54
$(6, 1.2, 0.9)$	79.23	75.19	76.99	80.89	75.34	78.03	81.05	80.29	**81.11**	79.83	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**	72.83	67.65
$(6, 1.4, 0.9)$	71.58	71.27	70.47	73.98	72.78	71.10	75.51	76.04	75.77	**76.94**	72.05
$(6, 1.4, 0.7)$	76.93	70.68	75.12	78.84	73.03	77.96	80.83	80.22	81.01	**82.19**	77.09
$(8, 1.0, 0.9)$	75.93	67.21	72.84	77.58	68.68	75.15	79.13	78.41	79.40	**81.11**	77.27
$(8, 1.0, 0.7)$	73.78	69.07	72.72	75.65	69.79	72.71	78.15	77.01	**78.64**	77.73	75.96
$(8, 1.2, 0.9)$	75.27	66.18	73.88	78.14	69.27	75.91	78.23	77.84	79.29	**79.83**	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**	72.27	68.41
$(8, 1.4, 0.9)$	75.66	68.16	74.85	77.78	70.08	75.77	81.04	79.34	80.41	**81.29**	76.88
$(8, 1.4, 0.7)$	72.94	68.66	73.21	76.44	70.21	73.88	78.26	76.56	**78.33**	78.16	74.42
$(10, 1.0, 0.9)$	67.04	65.75	62.46	68.81	67.20	64.06	70.53	70.43	**71.54**	66.73	65.28
$(10, 1.0, 0.7)$	73.32	66.82	71.09	74.80	68.76	72.32	**77.11**	74.66	76.79	75.56	73.26
$(10, 1.2, 0.9)$	68.61	62.56	66.24	70.72	63.42	67.95	73.89	71.43	**74.33**	71.35	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**	71.52	68.76
$(10, 1.4, 0.9)$	76.34	62.40	73.95	78.76	64.47	76.14	80.92	76.82	80.46	**82.37**	76.96
$(10, 1.4, 0.7)$	74.57	67.80	74.84	75.74	68.67	75.50	80.51	77.52	80.34	**81.49**	74.13
Avg.	73.71	68.29	71.82	75.60	69.78	73.31	77.43	76.41	77.90	77.47	73.65

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 and twelfth columns, we focus on the performance of GUB and 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 $100 \times \frac{\mathsf{Rev}}{\mathsf{UB}}$. We use the upper bound obtained from the dynamic program in (12), along with the revenue allocations provided by an optimal dual solution to problem (13). In each row, we indicate the best performing policy by using bold typeface.

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, LIN5, POL5, and LP5 improve the performance of LIN1, POL1, and LP1, respectively, by 2.56%, 2.18%, and 2.07%. The performance of LIN5 is still noticeably better than that of POL5 and LP5. When we use rollout, on average, LINR, POLR, and LPR improve the performance of LIN5, POL5, and LP5, respectively, by 2.42%, 9.50%, and 6.27%. Noting the performance of improvement of POLR over POL5, even if the static policy on hand is not a superior policy, performing rollout on the static policy can dramatically improve the static policy. Irrespective of whether the static policy is obtained by constructing value function approximations or by using a linear programming approximation, our rollout approach can be quite effective in obtaining even better policies. The performance of GUB is competitive to the rollout policies, even though we do not have a performance guarantee for GUB. Computation of the value function approximations for GUB uses ideas similar to those used for rolling out a static policy. The performance of DEC lags behind all rollout policies, which is encouraging, as dynamic programming decomposition is known to be one of the strongest benchmarks for revenue management problems prior to our work. In Appendix I, we compare our policies with the policy in Baek and Ma (2021). On average, LINR, POLR, and LPR improve on the policy in Baek and Ma (2021), respectively, by 22.91%, 21.29%, and 23.66%.

Another useful feature of rollout policies is that we can perform ensemble rollout on a collection of static policies. In particular, letting $\{\boldsymbol{\mu}_k : k = 1, \ldots, K\}$ be a collection of K static policies, we define the value function approximations $\{\hat{J}^q_{\max} : q \in \mathcal{Q}\}$ as $\hat{J}^q_{\max}(\boldsymbol{x}) = \max_{k=1,\ldots,K} J^q_{\boldsymbol{\mu}_k}(\boldsymbol{x})$, where $\{J^q_{\boldsymbol{\mu}_k} : q \in \mathcal{Q}\}$ are the value functions of the static policy $\boldsymbol{\mu}^k$ computed through the dynamic program in (2). In this case, we can show that the total expected revenue from the greedy policy with respect to the value function approximations $\{\hat{J}^q_{\max} : q \in \mathcal{Q}\}$ is at least as large as the total expected revenue from each of the static policies $\{\boldsymbol{\mu}_k : k = 1, \ldots, K\}$; see Example 2.3.2 in Bertsekas (2012). Thus, we can use ensemble rollout to obtain a policy that is at least as good as all of the static policies on hand. We performed such an ensemble rollout on the collection of the resource based static policy, itinerary based static policy, and static policy from problem (11). In all of our test problems, the performance of the ensemble rollout policy was statistically indistinguishable from the best performing 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 $\mathsf{D}_{\max} = 10$, 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. In Table 1, we give the gap between the upper bounds from our approach and the linear program in (11), rather than the absolute values of the upper bounds. Similarly, in Table 2, we give the gap between the total expected revenues of the benchmarks and the upper bound. In Appendix J, we give the absolute values for all upper bounds and total expected revenues obtained by all benchmarks. Lastly, we intuitively interpreted the fraction $\frac{\mathbb{1}_{\{\ell \in [s,f]\}}}{f-s+1}$ in (7) as a way of spreading the net revenue from a booking request for interval $[s, f]$ evenly over each day $\ell \in \{s, \ldots, f\}$. In Appendix K, we test a heuristic modification of LIN1, LIN5 and LINR that explores other ways of spreading the net revenue from a booking request.

7.2 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: In our data, we have access to the bookings 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. There are 119 days between May 13 and September 8, which is to say that the customers arrive over a selling horizon of 119 days. We divide each day into 10 time periods. Dividing each day into 10 time periods is enough to ensure that there is at most one customer arrival at each time period. In this case, the set of time periods is $\mathcal{Q} = \{1, \ldots, 1190\}$. For each $q \in \mathcal{Q}$, we use $\mathsf{day}(q)$ 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 $\mathcal{T} = \{1, \ldots, 153\}$ to index the days of stay. For each $\ell \in \mathcal{T}$, we use $\overline{\mathsf{day}}(\ell)$ to denote the calendar date corresponding to day ℓ. We use the following approach to estimate the parameters of our model.

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 $\lambda^q_{[s,f]} = \theta_{\overline{\mathsf{day}}(s) - \mathsf{day}(q)} \times \eta_{f-s+1}$. 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 $k = 1, \ldots, 171$. Noting that the dataset contains only 157 bookings, we divide the interval $\{1, \ldots, 171\}$ 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 $\{\theta_k : k = 1, \ldots, 171\}$, we impose the constraint that $\sum_{k=1}^{171} \theta_k \le 1$, in which case, we have $\sum_{[s,f] \in \mathcal{F}} \lambda^q_{[s,f]} \le 1$ 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

Load Fact.	Gap in Bounds	Total Exp. Revenue										
		LIN1	POL1	LP1	LIN5	POL5	LP5	LINR	POLR	LPR	GUB	DEC
0.8	12.25	81.50	83.69	81.41	82.59	83.33	82.05	83.96	**84.43**	83.00	83.26	82.10
1.0	11.13	84.55	86.96	84.55	85.25	86.77	84.79	86.91	**87.26**	85.88	86.19	85.12
1.2	10.10	85.76	87.73	85.69	87.34	87.75	86.69	88.18	**88.70**	87.16	88.29	86.43
1.4	8.95	86.96	88.63	87.11	88.25	88.76	87.22	89.38	89.36	88.31	**89.70**	87.34
1.6	8.15	87.19	89.35	87.47	88.16	89.44	87.58	**90.18**	89.69	88.85	90.14	87.40
Avg.	10.11	85.19	87.27	85.25	86.32	87.21	85.67	87.72	87.89	86.64	87.52	85.68

Table 3 Computational results for the boutique hotel datasets.

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 L, we give the details of our estimation procedure.

Computational Results: To broaden our experimental setup, we scaled the arrival probabilities $\{\lambda^q_{[s,f]} : [s,f] \in \mathcal{F},\ q \in \mathcal{Q}\}$ by a constant to obtain test problems with load factors taking values in $\{0.8, 1.0, 1.2, 1.4, 1.6\}$. 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 the dynamic program in (12) and the linear programming approximation in (11). 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 3.18%. Thus, our rollout approach continues to provide effective policies for the datasets based on actual boutique hotel bookings. The performance of GUB is competitive to that of the rollout policies.

8. Conclusion

We developed policies with performance guarantees by exploiting two features of the underlying revenue management problem. First, each resource is unique, having a single unit of capacity. Second, customers book resources for consecutive days. We hope that our study inspires other work to leverage features of the underlying problem to come up with policies with performance guarantees. In our computational experiments, performing rollout dramatically improved the performance of a static policy on hand and our rollout policies compared very favorably with the benchmarks. Moreover, the upper bounds on the optimal total expected revenue provided by our approach significantly tightened those from a linear programming approximation. Thus, our work provided not only stronger policies but also tighter upper bounds. Nevertheless, there is a remaining gap between the total expected revenues of our policies and the upper bounds in our test problems. Thus, one question is whether the performance of the policies can be improved or the upper bounds can be tightened or both. This question is difficult to answer and it is not unique to us, repeatedly arising in other settings when the optimal policy is difficult to compute and one makes comparisons only with an upper bound. In Appendix M, to partially answer this question, we consider problem instances with a single resource, so that $N = 1$. Even with a single resource, the state variable in (1) has T dimensions, but building on the approach in Section 6, we can efficiently solve this dynamic program by keeping track of the availability of the resource over uninterrupted intervals of days. In our results in Appendix M, the performance of the rollout policies is within a fraction of a percent of the optimal total expected revenue. Naturally, this finding does not ascertain that the performance of our rollout policies is within a fraction of a percent of the optimal when there is more than one resource, but it is comforting to verify that the rollout policies are essentially optimal for a case where we can compute the optimal policy.

There are several directions for research. In our model, a customer makes a booking request for a particular interval of use. 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 allows the customer to be flexible for her interval of use. Also, we focused on the total expected revenue over a finite planning horizon. It is common to use a finite planning horizon in hotel revenue management models, possibly because of the difficulty in estimating demand far into the future. An interesting avenue for research is to focus on the average revenue per time period over an infinite planning horizon. It is not clear to us how to incorporate the average revenue and bias terms into the recursions in (7) and (9). Another direction for research is to extend our results to the case where each resource has multiple units of capacity. The dynamic program that we use to compute the value functions of a static policy and the recursion that we use to construct the resource based static policy use the fact that each resource has unit capacity. Extensions to multiple units of capacity is nontrivial. One can heuristically use our policies in the presence of multiple units of capacity, where we deal with C units of capacity for a resource by creating C copies of the resource, each copy having one unit of capacity. In Appendix N, we experiment with such an extension. This extension is a heuristic. In particular, consider, for example, the case with three units of a resource so that we create three copies. The way we construct our value function approximations does not capture the fact that if the first copy is booked for days 1 and 2, the second copy is booked for days 2 and 3, and the third copy is booked for days 3 and 4, then we can still serve a booking request for days 1, 2, 3 and 4, by shifting the booking for days 3 and 4 from the third copy to the first copy, opening up the third copy. Although the way we construct our value function approximations does not capture this shifting possibility, when implementing the policy, we can heuristically incorporate such shifting possibilities. Extending our work to multiple units of capacity, while maintaining our rollout results, appears to be not straightforward and a full treatment of this case is naturally beyond our scope. Lastly, the performance guarantee of $\frac{1}{1+\|\mathcal{C}\|}$ 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.

Acknowledgements. We thank the department editor, associate editor and two anonymous referees whose comments significantly improved the paper in many ways. We gratefully appreciate that the benchmark policy `GUB` in Section 7 was suggested by the associate editor.

References

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

Adelman, Daniel, Adam J. Mersereau. 2008. Relaxations of weakly coupled stochastic dynamic programs. *Operations Research* **56**(3) 712–727.

Aouad, A., V. Farias, R. Levi. 2020. Assortment optimization under consider-then-choose choice models. *Management Science* (forthcoming).

Baek, J., W. Ma. 2021. 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. 4$^{\text{th}}$ ed. Athena Scientific, Belmont, MA.

Bertsekas, D. P. 2017. *Dynamic Programming and Optimal Control*, vol. 1. 4$^{\text{th}}$ 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.

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.

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

Villa Mahal. 2020. Villa Mahal Hotel website. URL `https://www.villamahal.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

Paat Rusmevichientong[1], Mika Sumida[1], Huseyin Topaloglu[2], Yicheng Bai[2]
[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

September 23, 2022

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 particular, we show that if the net revenue contribution from making a sale for a resource is nonpositive, then there exists an assortment that maximizes the net expected revenue from a customer without offering this resource. The proof of the lemma strictly uses the fact that the choice probabilities of the resources in an assortment decrease as we add more resources into the assortment; that is, $\phi_i^q(S\cup\{j\}) \le \phi_i^q(S)$ for all $i \in S$ and $j \notin S$. For fixed net revenue contributions $\{p_i : i \in \mathcal{N}\}$, we focus on the problem

$$\max_{S\subseteq \mathcal{N}} \left\{ \sum_{i\in\mathcal{N}} \phi_i^q(S)\, p_i \right\}. \tag{16}$$

Lemma A.1 (Offering Only Positive Contributions) *There exists an optimal solution S^* to problem* (16) *such that $S^* \subseteq \{i\in\mathcal{N} : p_i > 0\}$.*

Proof: Letting S^* be an optimal solution to problem (16), we define the assortment $\hat{S} \subseteq S^*$ as $\hat{S} = \{i \in S^* : p_i > 0\}$. By the assumption that $\phi_i^q(S\cup\{j\}) \le \phi_i^q(S)$ for all $i\in S$ and $j \notin S$, we have $\phi_i^q(S^*) \le \phi_i^q(\hat{S})$ for all $i \in \hat{S}$. In this case, using the fact that $p_i \le 0$ 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

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

Thus, the chain of inequalities above shows that $\hat{S}$ is also an optimal solution to problem (16). Furthermore, we have $\hat{S} \subseteq \{i\in\mathcal{N} : p_i > 0\}$. ■

Appendix B: Performance Guarantee for Resource Based Static Policy

We give the proof of Theorem 4.1. The proof 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.

B.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 $\boldsymbol{x}$.

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

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

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 $\boldsymbol{x}\in\{0,1\}^{N\times T}$. The next lemma shows that the contribution of each resource to the objective value in (6) is nonnegative.

Lemma B.2 (Nonnegative Contribution of Each Resource) *For each $i\in\mathcal{N}$, $[s,f]\in\mathcal{F}$, and $q\in\mathcal{Q}$, we have $\phi_i^q(A^q_{[s,f]})\big[r_{i,[s,f]}-\sum_{h=s}^{f}\eta^{q+1}_{i,h}\big]\geq 0$.*

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

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

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 $A^q_{[s,f]}$ 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 B.3 (Upper Bound on Net Expected Revenue) *Letting $\{A^q_{[s,f]} : [s,f] \in \mathcal{F},\ q \in \mathcal{Q}\}$ and $\{\eta^q_{i,\ell} : i \in \mathcal{N},\ \ell \in \mathcal{T},\ q \in \mathcal{Q}\}$ be computed through* (6) *and* (7)*, we have*

$$\sum_{[s,f]\in\mathcal{F}} \lambda^q_{[s,f]} \sum_{i\in\mathcal{N}} \phi^q_i\big(A^q_{[s,f]}\big) \frac{f-s}{f-s+1} \left[r_{i,[s,f]} - \sum_{h=s}^{f} \eta^{q+1}_{i,h} \right] \le \frac{\mathsf{D}_{\max}-1}{\mathsf{D}_{\max}} \big(\hat{J}^q_{\mathsf{L}}(\boldsymbol{e}) - \hat{J}^{q+1}_{\mathsf{L}}(\boldsymbol{e})\big).$$

Proof: Because $x/(x+1)$ is increasing in $x \in \mathbb{R}_+$ and $f-s+1 \le \mathsf{D}_{\max}$, we have $\frac{f-s}{f-s+1} \le \frac{\mathsf{D}_{\max}-1}{\mathsf{D}_{\max}}$. Noting that $\phi^q_i(A^q_{[s,f]})\big[r_{i,[s,f]} - \sum_{h=s}^{f} \eta^{q+1}_{i,h}\big] \ge 0$ for all $i \in \mathcal{N}$ and $[s,f] \in \mathcal{F}$ by Lemma B.2, we get

$$\begin{aligned}
&\sum_{[s,f]\in\mathcal{F}} \lambda^q_{[s,f]} \sum_{i\in\mathcal{N}} \phi^q_i\big(A^q_{[s,f]}\big)\ \frac{f-s}{f-s+1} \left[r_{i,[s,f]}\ - \sum_{h=s}^{f} \eta^{q+1}_{i,h} \right] \\
&\qquad \le\ \frac{\mathsf{D}_{\max}-1}{\mathsf{D}_{\max}} \sum_{i\in\mathcal{N}} \sum_{[s,f]\in\mathcal{F}} \lambda^q_{[s,f]}\ \phi^q_i(A^q_{[s,f]}) \left[r_{i,[s,f]}\ - \sum_{h=s}^{f} \eta^{q+1}_{i,h} \right] \\
&\qquad \overset{(a)}{=}\ \frac{\mathsf{D}_{\max}-1}{\mathsf{D}_{\max}} \sum_{i\in\mathcal{N}} \sum_{\ell\in\mathcal{T}} \sum_{[s,f]\in\mathcal{F}} \lambda^q_{[s,f]}\ \phi^q_i(A^q_{[s,f]})\, \frac{\mathbb{1}_{\{\ell\in[s,f]\}}}{f-s+1} \left[r_{i,[s,f]}\ - \sum_{h=s}^{f} \eta^{q+1}_{i,h} \right] \\
&\qquad \overset{(b)}{=}\ \frac{\mathsf{D}_{\max}-1}{\mathsf{D}_{\max}} \sum_{i\in\mathcal{N}} \sum_{\ell\in\mathcal{T}} (\eta^q_{i,\ell} - \eta^{q+1}_{i,\ell})\ \overset{(c)}{=}\ \frac{\mathsf{D}_{\max}-1}{\mathsf{D}_{\max}} \big(\hat{J}^q_{\mathsf{L}}(\boldsymbol{e}) - \hat{J}^{q+1}_{\mathsf{L}}(\boldsymbol{e})\big),
\end{aligned}$$

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 $\hat{J}^q_{\mathsf{L}}(\boldsymbol{x}) = \sum_{i\in\mathcal{N}} \sum_{\ell\in\mathcal{T}} \eta^q_{i,\ell}\, x_{i,\ell}$.

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

B.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 $\boldsymbol{x} \in \{0,1\}^{N\times T}$ and $q \in \mathcal{Q}$, the value function approximations $\{\widetilde{J}^q : q \in \mathcal{Q}\}$ satisfy

$$\widetilde{J}^q(\boldsymbol{x}) \ge \sum_{[s,f]\in\mathcal{F}} \lambda^q_{[s,f]} \max_{S\subseteq\mathcal{N}} \left\{ \sum_{i\in\mathcal{N}} \phi^q_i(S) \left(\prod_{\ell=s}^{f} x_{i,\ell} \right) \left[r_{i,[s,f]} + \widetilde{J}^{q+1}(\boldsymbol{x} - \boldsymbol{e}_{i,[s,f]}) - \widetilde{J}^{q+1}(\boldsymbol{x}) \right] \right\} + \widetilde{J}^{q+1}(\boldsymbol{x}), \quad (17)$$

then we have $\widetilde{J}^q(\boldsymbol{x}) \ge J^q(\boldsymbol{x})$ for all $\boldsymbol{x} \in \{0,1\}^{N\times T}$ 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 $\{\widetilde{J}_q : q \in \mathcal{Q}\}$ 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 $2\hat{J}^1_{\mathsf{L}}(\boldsymbol{e})$ 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 B.4 (Upper Bound on Optimal Performance) *Noting that the optimal total expected revenue is $J^1(\boldsymbol{e})$, we have $J^1(\boldsymbol{e}) \le 2\,\hat{J}^1_{\mathsf{L}}(\boldsymbol{e})$.*

Proof: Using our linear value function approximations $\{\hat{J}^q_{\mathsf{L}} : q \in \mathcal{Q}\}$ computed through (6) and (7), let $\widetilde{J}^q(\boldsymbol{x}) = \hat{J}^q_{\mathsf{L}}(\boldsymbol{e}) + \hat{J}^q_{\mathsf{L}}(\boldsymbol{x})$. We show that $\{\widetilde{J}^q : q \in \mathcal{Q}\}$ satisfies (17). In particular, we have

$$
\begin{aligned}
&\sum_{[s,f]\in\mathcal{F}} \lambda^q_{[s,f]} \max_{S\subseteq\mathcal{N}} \left\{ \sum_{i\in\mathcal{N}} \phi^q_i(S) \left(\prod_{\ell=s}^{f} x_{i,\ell} \right) \left[r_{i,[s,f]} + \widetilde{J}^{q+1}(\boldsymbol{x}) - \widetilde{J}^{q+1}(\boldsymbol{x} - \boldsymbol{e}_{i,[s,f]}) \right] \right\} \\
&\qquad = \sum_{[s,f]\in\mathcal{F}} \lambda^q_{[s,f]} \max_{S\subseteq\mathcal{N}} \left\{ \sum_{i\in\mathcal{N}} \phi^q_i(S) \left(\prod_{\ell=s}^{f} x_{i,\ell} \right) \left[r_{i,[s,f]} - \sum_{h=s}^{f} \eta^{q+1}_{i,h} \right] \right\} \\
&\qquad \overset{(a)}{\le} \sum_{[s,f]\in\mathcal{F}} \lambda^q_{[s,f]} \max_{S\subseteq\mathcal{N}} \left\{ \sum_{i\in\mathcal{N}} \phi^q_i(S) \left[r_{i,[s,f]} - \sum_{h=s}^{f} \eta^{q+1}_{i,h} \right] \right\} \\
&\qquad \overset{(b)}{=} \sum_{[s,f]\in\mathcal{F}} \lambda^q_{[s,f]} \sum_{i\in\mathcal{N}} \phi^q_i(A^q_{[s,f]}) \left[r_{i,[s,f]} - \sum_{h=s}^{f} \eta^{q+1}_{i,h} \right] \\
&\qquad \overset{(c)}{=} \sum_{i\in\mathcal{N}} \sum_{[s,f]\in\mathcal{F}} \lambda^q_{[s,f]}\, \phi^q_i(A^q_{[s,f]})\, \frac{\sum_{\ell\in\mathcal{T}} \mathbb{1}_{\{\ell\in[s,f]\}}}{f-s+1} \left[r_{i,[s,f]} - \sum_{h=s}^{f} \eta^{q+1}_{i,h} \right] \\
&\qquad = \sum_{i\in\mathcal{N}} \sum_{\ell\in\mathcal{T}} \sum_{[s,f]\in\mathcal{F}} \lambda^q_{[s,f]}\, \phi^q_i(A^q_{[s,f]})\, \frac{\mathbb{1}_{\{\ell\in[s,f]\}}}{f-s+1} \left[r_{i,[s,f]} - \sum_{h=s}^{f} \eta^{q+1}_{i,h} \right] \\
&\qquad \overset{(d)}{=} \sum_{i\in\mathcal{N}} \sum_{\ell\in\mathcal{T}} \left(\eta^q_{i,\ell} - \eta^{q+1}_{i,\ell} \right) \overset{(e)}{\le} \sum_{i\in\mathcal{N}} \sum_{\ell\in\mathcal{T}} \left(\eta^q_{i,\ell} - \eta^{q+1}_{i,\ell} \right) (1 + x_{i,\ell}) = \widetilde{J}^q(\boldsymbol{x}) - \widetilde{J}^{q+1}(\boldsymbol{x}).
\end{aligned}
$$

The inequality (a) holds because Lemma A.1 in Appendix A implies that if $r_{i,[s,f]} - \sum_{h=s}^{f} \eta^{q+1}_{i,h} \le 0$, then there exists an optimal solution to the problem on the left side of (a) such that resource i is not offered. So, if $r_{i,[s,f]} - \sum_{h=s}^{f} \eta^{q+1}_{i,h} \le 0$, then we can drop resource i from consideration in this problem, but if $r_{i,[s,f]} - \sum_{h=s}^{f} \eta^{q+1}_{i,h} > 0$, then we have $\left(\prod_{\ell=s}^{f} x_{i,\ell}\right) [r_{i,[s,f]} - \sum_{h=s}^{f} \eta^{q+1}_{i,h}] \le r_{i,[s,f]} - \sum_{h=s}^{f} \eta^{q+1}_{i,h}$. Moreover, (b) follows from the definition of $A^q_{[s,f]}$ 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 $\eta^q_{i,\ell}$ in (7). Lastly, (e) holds because we have $\phi^q_i(A^q_{[s,f]})\big[r_{i,[s,f]} - \sum_{h=s}^{f} \eta^{q+1}_{i,h}\big] \ge 0$ by Lemma B.2, which implies that $\eta^q_{i,\ell} \ge \eta^{q+1}_{i,\ell}$ by (7).

The chain of inequalities above shows that $\{\widetilde{J}^q : q \in \mathcal{Q}\}$ satisfies (17). Therefore, we have $\widetilde{J}^q(\boldsymbol{x}) \ge J^q(\boldsymbol{x})$ for all $\boldsymbol{x} \in \{0,1\}^{N\times T}$ and $q \in \mathcal{Q}$, so $J^1(\boldsymbol{e}) \le \widetilde{J}^1(\boldsymbol{e}) = 2\,\hat{J}^1_{\mathsf{L}}(\boldsymbol{e})$. ∎

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.

B.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 $U^q_{\mathsf{L}}(\boldsymbol{x})$ be the total expected revenue obtained by the resource based static policy over time periods $\{q,\ldots,Q\}$ given that the state of the system at time period q is $\boldsymbol{x}$. We can compute the value functions $\{U_{\mathsf{L}}^q : q \in \mathcal{Q}\}$ through the dynamic program

$$U_{\mathsf{L}}^q(\boldsymbol{x}) = \sum_{[s,f]\in\mathcal{F}} \lambda_{[s,f]}^q \left\{ \sum_{i\in\mathcal{N}} \phi_i^q(A_{[s,f]}^q) \left(\prod_{\ell=s}^f x_{i,\ell} \right) \left[r_{i,[s,f]} + U_{\mathsf{L}}^{q+1}(\boldsymbol{x} - \boldsymbol{e}_{i,[s,f]}) - U_{\mathsf{L}}^{q+1}(\boldsymbol{x}) \right] \right\} + U_{\mathsf{L}}^{q+1}(\boldsymbol{x}), \quad (18)$$

with the boundary condition that $U_{\mathsf{L}}^{Q+1} = 0$. 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 $\{\hat{J}_{\mathsf{L}}^q : q \in \mathcal{Q}\}$ to give a lower bound on the performance of the resource based static policy. Lemma B.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 B.5 (Lower Bound on Policy Performance) *Letting the value functions* $\{U_{\mathsf{L}}^q : q \in \mathcal{Q}\}$ *be computed through* (18)*, for each* $\boldsymbol{x} \in \{0,1\}^{N\times T}$ *and* $q \in \mathcal{Q}$*, we have*

$$U_{\mathsf{L}}^q(\boldsymbol{x}) \;\geq\; \hat{J}_{\mathsf{L}}^q(\boldsymbol{x}) \;-\; \frac{\mathsf{D}_{\max} - 1}{\mathsf{D}_{\max}}\, \hat{J}_{\mathsf{L}}^q(\boldsymbol{e}).$$

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 B.1 and $\phi_i^q(A_{[s,f]}^q)\big[r_{i,[s,f]} - \sum_{h=s}^f \eta_{i,h}^{q+1}\big] \geq 0$ by Lemma B.2, we have

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

where (a) follows by arranging the terms on the left side of (a) and using Lemma B.3, (b) uses (7), and (c) uses the fact that $\hat{J}_{\mathsf{L}}^q(\boldsymbol{x}) = \sum_{i\in\mathcal{N}} \sum_{\ell\in\mathcal{T}} \eta_{i,\ell}\, x_{i,\ell}$.

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 $U_{\mathsf{L}}^{Q+1} = 0 = \hat{J}_{\mathsf{L}}^{Q+1}$, 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 $U_{\mathsf{L}}^{q+1}(\boldsymbol{x})$ on the right side of (18) 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 $\sum_{[s,f]\in\mathcal{F}}\lambda_{[s,f]}^q\le 1$ and $\sum_{i\in\mathcal{N}}\phi_i^q(A_{[s,f]}^q)\le 1$, this coefficient is nonnegative. Letting $\alpha^q=\frac{\mathsf{D}_{\max}-1}{\mathsf{D}_{\max}}\hat{J}_{\mathsf{L}}^q(\boldsymbol{e})$ for notational brevity, by the induction assumption, we have $U_{\mathsf{L}}^{q+1}(\boldsymbol{x})\ge\hat{J}_{\mathsf{L}}^{q+1}(\boldsymbol{x})-\alpha^{q+1}$ for all $\boldsymbol{x}\in\{0,1\}^{N\times T}$. In this case, replacing $U_{\mathsf{L}}^{q+1}(\boldsymbol{x})$ and $U_{\mathsf{L}}^{q+1}(\boldsymbol{x}-\boldsymbol{e}_{i,[s,f]})$ on the right side of (18) with their corresponding lower bounds $\hat{J}_{\mathsf{L}}^{q+1}(\boldsymbol{x})-\alpha^{q+1}$ and $\hat{J}_{\mathsf{L}}^{q+1}(\boldsymbol{x}-\boldsymbol{e}_{i,[s,f]})-\alpha^{q+1}$, respectively, the right side of (18) gets smaller, so we get the chain of inequalities

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

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

Finally, we can use Propositions B.4 and B.5 to give a proof for Theorem 4.1.

Proof of Theorem 4.1:

By Proposition B.4, we have $\hat{J}_{\mathsf{L}}^1(\boldsymbol{e})\ge\frac{1}{2}J^1(\boldsymbol{e})$. Using Proposition B.5 with $\boldsymbol{x}=\boldsymbol{e}$ and $q=1$, we have $U_{\mathsf{L}}^1(\boldsymbol{e})\ge\hat{J}_{\mathsf{L}}^1(\boldsymbol{e})-\frac{\mathsf{D}_{\max}-1}{\mathsf{D}_{\max}}\hat{J}_{\mathsf{L}}^1(\boldsymbol{e})=\frac{1}{\mathsf{D}_{\max}}\hat{J}_{\mathsf{L}}^1(\boldsymbol{e})$, so we get $U_{\mathsf{L}}^1(\boldsymbol{e})\ge\frac{1}{\mathsf{D}_{\max}}\hat{J}_{\mathsf{L}}^1(\boldsymbol{e})\ge\frac{1}{2\,\mathsf{D}_{\max}}J^1(\boldsymbol{e})$. ■

Appendix C: Performance Guarantee for Itinerary Based Static Policy

In this section, we give a proof for Theorem 5.1. We will use two preliminary lemmas. The next lemma is the analogue of Lemma B.2. Its proof is identical to that of Lemma B.2 and omitted.

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

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

Proof: Using the previous lemma, by (9), we have $\gamma^q_{i,[s,f]} \geq 0$ for all $i \in \mathcal{N}$ and $[s,f] \in \mathcal{F}$. Using the fact that $\hat{J}^q_{\mathsf{P}}(\boldsymbol{x}) = \sum_{i\in\mathcal{N}} \sum_{[s,f]\in\mathcal{F}} \gamma^q_{i,[s,f]} \prod_{\ell=s}^f x_{i,\ell}$, the difference $\hat{J}^q_{\mathsf{P}}(\boldsymbol{x}) - \hat{J}^q_{\mathsf{P}}(\boldsymbol{x} - \boldsymbol{e}_{i,[s,f]})$ is

$$\begin{aligned}
\hat{J}^q_{\mathsf{P}}(\boldsymbol{x}) - \hat{J}^q_{\mathsf{P}}(\boldsymbol{x} - \boldsymbol{e}_{i,[s,f]}) &= \sum_{[a,b]\in\mathcal{F}} \gamma^q_{i,[a,b]}\, \mathbb{1}_{\{[a,b]\cap[s,f]\neq\varnothing\}} \left(\prod_{\ell=a}^{b} x_{i,\ell}\right) \\
&\leq \sum_{[a,b]\in\mathcal{F}} \gamma^q_{i,[a,b]}\, \mathbb{1}_{\{[a,b]\cap[s,f]\neq\varnothing\}} \;\leq\; \sum_{[a,b]\in\mathcal{F}} \big|\,[s,f]\cap C_{[a,b]}\,\big|\, \gamma^q_{i,[a,b]},
\end{aligned}$$

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

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

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

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

$$\begin{aligned}
&\sum_{[s,f]\in\mathcal{F}} \lambda^q_{[s,f]} \max_{S\subseteq\mathcal{N}} \left\{ \sum_{i\in\mathcal{N}} \phi^q_i(S) \left(\prod_{\ell=s}^{f} x_{i,\ell}\right) \Big[r_{i,[s,f]} + \hat{V}^{q+1}_{\mathsf{P}}(\boldsymbol{x}-\boldsymbol{e}_{i,[s,f]}) - \hat{V}^{q+1}_{\mathsf{P}}(\boldsymbol{x}) \Big] \right\} \\
&\quad= \sum_{[s,f]\in\mathcal{F}} \lambda^q_{[s,f]} \max_{S\subseteq\mathcal{N}} \left\{ \sum_{i\in\mathcal{N}} \phi^q_i(S) \left(\prod_{\ell=s}^{f} x_{i,\ell}\right) \Big[r_{i,[s,f]} - \sum_{h=s}^{f} \beta^{q+1}_{i,h} \Big] \right\} \\
&\quad\overset{(a)}{\leq} \sum_{[s,f]\in\mathcal{F}} \lambda^q_{[s,f]} \max_{S\subseteq\mathcal{N}} \left\{ \sum_{i\in\mathcal{N}} \phi^q_i(S) \Big[r_{i,[s,f]} - \sum_{h=s}^{f} \beta^{q+1}_{i,h} \Big] \right\} \\
&\quad\overset{(b)}{=} \sum_{[s,f]\in\mathcal{F}} \lambda^q_{[s,f]} \max_{S\subseteq\mathcal{N}} \left\{ \sum_{i\in\mathcal{N}} \phi^q_i(S) \Big[r_{i,[s,f]} - \sum_{[a,b]\in\mathcal{F}} \Big| C_{[a,b]}\cap[s,f] \Big| \gamma^{q+1}_{i,[a,b]} \Big] \right\} \\
&\quad\overset{(c)}{=} \sum_{[s,f]\in\mathcal{F}} \lambda^q_{[s,f]} \sum_{i\in\mathcal{N}} \phi^q_i(B^q_{[s,f]}) \Big[r_{i,[s,f]} - \sum_{[a,b]\in\mathcal{F}} \Big| C_{[a,b]}\cap[s,f] \Big| \gamma^{q+1}_{i,[a,b]} \Big] \\
&\quad\overset{(d)}{=} \sum_{i\in\mathcal{N}} \sum_{[s,f]\in\mathcal{F}} (\gamma^q_{i,[s,f]} - \gamma^{q+1}_{i,[s,f]}) \;=\; \hat{J}^q_{\mathsf{P}}(\boldsymbol{e}) - \hat{J}^{q+1}_{\mathsf{P}}(\boldsymbol{e}) \\
&\quad\overset{(e)}{\leq} \hat{J}^q_{\mathsf{P}}(\boldsymbol{e}) - \hat{J}^{q+1}_{\mathsf{P}}(\boldsymbol{e}) + \sum_{i\in\mathcal{N}}\sum_{\ell\in\mathcal{T}} (\beta^q_{i,\ell} - \beta^{q+1}_{i,\ell})\, x_{i,\ell} \;=\; \hat{V}^q_{\mathsf{P}}(\boldsymbol{x}) - \hat{V}^{q+1}_{\mathsf{P}}(\boldsymbol{x}).
\end{aligned}$$

Here, (a) follows from the same argument that we use to obtain the inequality (a) in the proof of Proposition B.4. In (b), we use the definition of $\beta^q_{i,\ell}$ and use the interchange of sums $\sum_{h=s}^{f}\sum_{[a,b]\in\mathcal{F}} \mathbb{1}_{\{h\in C_{[a,b]}\}} = \sum_{[a,b]\in\mathcal{F}}\sum_{h=s}^{f} \mathbb{1}_{\{h\in C_{[a,b]}\}} = \sum_{[a,b]\in\mathcal{F}} |C_{[a,b]}\cap[s,f]|$. Also, (c) follows from (8), and (d) follows from (9). To see that (e) holds, note that Lemma C.1, along with (9), implies that $\gamma^q_{i,[s,f]} \geq \gamma^{q+1}_{i,[s,f]}$, in which case, we also have $\beta^q_{i,\ell} \geq \beta^{q+1}_{i,\ell}$. The chain of inequalities above shows that the value function approximations $\{\hat{V}_{\mathsf{P}}^q : q \in \mathcal{Q}\}$ satisfy (17), in which case, by the discussion that follows (17), we have $\hat{V}_{\mathsf{P}}^q(\boldsymbol{x}) \geq J^q(\boldsymbol{x})$ for all $\boldsymbol{x} \in \{0,1\}^{N\times T}$ and $q \in \mathcal{Q}$. Thus, we have

$$\begin{aligned} J^1(\boldsymbol{e}) \;\leq\; \hat{V}_{\mathsf{P}}^1(\boldsymbol{e}) \;=\; \hat{J}_{\mathsf{P}}^1(\boldsymbol{e}) + \sum_{i\in\mathcal{N}}\sum_{\ell\in\mathcal{T}} \beta_{i,\ell}^1 \;=\; \hat{J}_{\mathsf{P}}^1(\boldsymbol{e}) + \sum_{i\in\mathcal{N}}\sum_{[a,b]\in\mathcal{F}}\sum_{\ell\in\mathcal{T}} \mathbb{1}_{\{\ell\in C_{[a,b]}\}}\, \gamma_{i,[a,b]}^1 \\ =\; \hat{J}_{\mathsf{P}}^1(\boldsymbol{e}) + \sum_{i\in\mathcal{N}}\sum_{[a,b]\in\mathcal{F}} \left|C_{[a,b]}\right| \gamma_{i,[a,b]}^1 \;\leq\; \hat{J}_{\mathsf{P}}^1(\boldsymbol{e}) + \|\mathcal{C}\| \sum_{i\in\mathcal{N}}\sum_{[a,b]\in\mathcal{F}} \gamma_{i,[a,b]}^1 \;=\; (1+\|\mathcal{C}\|)\, \hat{J}_{\mathsf{P}}^1(\boldsymbol{e}), \end{aligned}$$

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 $\hat{J}_{\mathsf{P}}^q(\boldsymbol{e}) = \sum_{i\in\mathcal{N}}\sum_{[a,b]\in\mathcal{F}} \gamma_{i,[a,b]}^q$. ■

Let $U_{\mathsf{P}}^q(\boldsymbol{x})$ be the total expected revenue obtained by the itinerary based static policy over time periods $\{q,\ldots,Q\}$ given that the state of the system at time period q is $\boldsymbol{x}$. We can compute the value functions $\{U_{\mathsf{P}}^q : q \in \mathcal{Q}\}$ through the dynamic program in (18) 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 C.4 (Lower Bound on Policy Performance) *Letting $\{U_{\mathsf{P}}^q : q \in \mathcal{Q}\}$ be the value functions of the itinerary based static policy, for each $\boldsymbol{x} \in \{0,1\}^{N\times T}$ and $q \in \mathcal{Q}$, $U_{\mathsf{P}}^q(\boldsymbol{x}) \;\geq\; \hat{J}_{\mathsf{P}}^q(\boldsymbol{x})$.*

Proof: We show the result by using induction over the time periods. At time period $Q+1$, since we have $U_{\mathsf{P}}^{Q+1} = 0 = \hat{J}_{\mathsf{P}}^{Q+1}$, 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, $U_{\mathsf{P}}^{q+1}(\boldsymbol{x})$ and $U_{\mathsf{P}}^{q+1}(\boldsymbol{x} - \boldsymbol{e}_{i,[s,f]})$ are, respectively, lower bounded by $\hat{J}_{\mathsf{P}}^{q+1}(\boldsymbol{x})$ and $\hat{J}_{\mathsf{P}}^{q+1}(\boldsymbol{x} - \boldsymbol{e}_{i,[s,f]})$, so by (18), we get

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

where (a) follows from Lemma C.2, (b) holds by (9), and (c) holds by the definition of $\hat{J}_{\mathsf{P}}^q(\boldsymbol{x})$. The chain of inequalities above establishes the result. ■

We can use Propositions C.3 and C.4 to give a proof for Theorem 5.1.

Proof of Theorem 5.1:

By Proposition C.3, we have $\hat{J}_{\mathsf{P}}^1(\boldsymbol{e}) \geq \frac{1}{1+\|\mathcal{C}\|} J^1(\boldsymbol{e})$. Using Proposition C.4 with $\boldsymbol{x} = \boldsymbol{e}$ and $q = 1$, we have $U_{\mathsf{P}}^1(\boldsymbol{e}) \geq \hat{J}_{\mathsf{P}}^1(\boldsymbol{e})$. So, we get $U_{\mathsf{P}}^1(\boldsymbol{e}) \geq \hat{J}_{\mathsf{P}}^1(\boldsymbol{e}) \geq \frac{1}{1+\|\mathcal{C}\|}\, J^1(\boldsymbol{e})$. ■

Appendix D: Tightness of the Performance Guarantee

We show that our performance guarantee for the itinerary based static policy is tight. Consider a problem instance with $N+1$ resources indexed by $\mathcal{N}=\{1,\ldots,N+1\}$, where we assume that $N \geq 2$. The resources are available for use during the days $\mathcal{T}=\{1,\ldots,T\}$. The set of possible intervals of use are $\mathcal{F}=\{[t,t]: t\in\mathcal{T}\}\cup\{[1,T]\}$. Thus, there are $T+1$ possible intervals of use. Each of the first T possible intervals of use occupies a resource for only one day, whereas the last possible interval of use occupies the resource for the whole interval $[1,T]$. The revenue from booking any resource i over the interval $[t,t]$ is $r_{i,[t,t]}=\frac{N-1}{N\,(N+1)}$, whereas the revenue from booking any resource i over the interval $[1,T]$ is $r_{i,[1,T]}=1$. Note that the revenues from the booking requests depend on the interval of use, but not on the specific resource that is used to serve the booking request. Because $N \geq 2$, we have $r_{i,[t,t]}>0$. The booking requests arrive over the time periods $\mathcal{Q}=\{1,\ldots,NT+1\}$. There is one customer arrival at each time period. For $t=1,\ldots,T$, at each of the time periods $\{(t-1)\,N+1,\ldots,tN\}$, we have a booking request for the interval $[t,t]$ with probability one, whereas at the last time period $NT+1$, we have a booking request for the interval $[1,T]$ with probability one. In other words, the booking requests for the interval $[t,t]$ arrive only at time periods $\{(t-1)\,N+1,\ldots,tN\}$. There is a single booking request for the interval $[1,T]$ and it arrives at the last time period. At each of the first NT time periods, given that we offer the assortment S of resources, the customer chooses each offered resource with an equal probability of $\frac{1}{|S|}$, never leaving the system without choosing any of the offered resources as along as a resource is offered to this customer. At the last time period $NT+1$, given that we offer the assortment S of resources, the customer chooses among the offered resources with an equal probability of $\frac{N}{1+N\,|S|}$, leaving the system without choosing any of the offered resources with a probability of $\frac{1}{1+N\,|S|}$. At each of the first NT time periods, the customer never leaves without choosing an offered resource, so offering a single resource ensures that the customer chooses this resource.

We compute the total expected revenue of a benchmark policy, which provides a lower bound on the optimal total expected revenue. The benchmark policy uses one of the first N resources for the customers arriving at the first NT time periods, leaving the last resource untouched for the customer arriving at the last time period. For $t=1,\ldots,T$, the customers arriving at each of the time periods $\{(t-1)\,N+1,\ldots,tN\}$ make a booking request for the interval $[t,t]$. Thus, there are N customers making a booking request for the interval $[t,t]$. The benchmark policy offers only one of the first N resources to each one of these customers. As discussed earlier, the customers arriving at one of the first NT time periods always chooses the resource offered to them. Thus, the benchmark policy uses the first N resources to serve the booking requests from the customers arriving at time periods $\{(t-1)\,N+1,\ldots,tN\}$, in which case, noting that the revenue from a booking request for the interval $[t,t]$ is $\frac{N-1}{N\,(N+1)}$, the benchmark policy obtains a total expected revenue of $N\frac{N-1}{N(N+1)}$ over time periods $\{(t-1)\,N+1,\ldots,tN\}$. Thus, the benchmark policy obtains a total expected revenue of $TN\frac{N-1}{N(N+1)}$ from the first NT customers. The last resource is left untouched for the customer arriving at the last time period $NT+1$. The benchmark policy offers only this resource to the customer at the last time period and the customer chooses the resource with probability $\frac{N}{1+N}$. The revenue from a booking request for the interval $[1,T]$ is one, so the benchmark policy obtains a total expected revenue of $\frac{N}{1+N}$ at the last time period. Thus, the total expected revenue of the benchmark policy is $T\,\frac{N-1}{N+1}+\frac{N}{1+N}$, which is arbitrarily close to $T+1$ as N gets large. Therefore, letting OPT be the optimal total expected revenue, $\mathsf{OPT}\geq T+1-\frac{1}{N}$ for large enough N.

The intersection preserving subset for the interval $[1,T]$ must be $C_{[1,T]}=\{1,\ldots,T\}$. If $t\notin C_{[1,T]}$ for some $t\in\mathcal{T}$, then we get $[1,T]\cap[t,t]\neq\varnothing$ and $C_{[1,T]}\cap[t,t]=\varnothing$, which violates the definition of an intersection preserving subset. The only intersection preserving subset for the interval $[t,t]$ is $C_{[t,t]}=\{t\}$. Thus, there is a unique collection $\mathcal{C}$ of intersection preserving subsets for our problem instance and its norm is T. Consider (8) for the interval $[s,f]=[1,T]$ at time period $q=NT+1$. We have $B^{NT+1}_{[1,T]}=\arg\max_{S\subseteq\mathcal{N}}\sum_{i\in\mathcal{N}}\mathbf{1}(i\in S)\frac{N}{1+N\,|S|}=\arg\max_{S\subseteq\mathcal{N}}\frac{N\,|S|}{1+N\,|S|}$. Because $\frac{N\,|S|}{1+N\,|S|}$ is increasing in $|S|$, we get $B^{NT+1}_{[1,T]}=\mathcal{N}$. Plugging the assortment $B^{NT+1}_{[1,T]}=\mathcal{N}$ into (9), noting that $\phi_i^{NT+1}(\mathcal{N})=\frac{N}{1+N\,(N+1)}$ and $|\mathcal{N}|=N+1$, we have $\gamma^{NT+1}_{i,[1,T]}=\frac{N}{1+N\,(N+1)}$. We get a booking request for the interval $[1,T]$ only at the last time period, so $\gamma^{q}_{i,[1,T]}=\frac{N}{1+N\,(N+1)}$ for all $q\in\mathcal{Q}$.

Consider time period tN. This time period is the last one at which we have a booking request for the interval $[t,t]$. Therefore, by (8), we have

$$
\begin{aligned}
B^{tN}_{[t,t]} &= \arg\max_{S\subseteq\mathcal{N}}\left\{\sum_{i\in\mathcal{N}}\phi_i^q(S)\Big[r_{i,[t,t]}\;-\;\gamma^{tN+1}_{i,[t,t]}-\gamma^{tN+1}_{i,[1,T]}\Big]\right\}.\\
&= \arg\max_{S\subseteq\mathcal{N}}\left\{\sum_{i\in\mathcal{N}}\mathbf{1}(S\neq\varnothing)\,\frac{1}{|S|}\left[\frac{N-1}{N\,(N+1)}\;-\;0-\frac{N}{1+N\,(N+1)}\right]\right\}\;=\;\varnothing,
\end{aligned}
$$

where the last equality holds because $\frac{N-1}{N\,(N+1)}-\frac{N}{1+N\,(N+1)}<0$, so any non-empty solution to the problem above yields a strictly negative objective value. Plugging the assortment $B^{tN}_{[t,t]}=\varnothing$ into (9) yields $\gamma^{tN}_{i,[t,t]}=0$. Moving backwards over the time periods and continuing in the same fashion, it follows that $B^{q}_{[t,t]}=\varnothing$ and $\gamma^{q}_{i,[t,t]}=0$ for all $i\in\mathcal{N}$ and $q\in\{(t-1)N+1,\ldots,tN\}$. Therefore, the itinerary based static policy offers the empty assortment of resources to all of the booking requests for a single day and offers the full assortment of resources to the booking request for the interval $[1,T]$ that arrives at the last time period. In this case, letting STA be the total expected revenue of the itinerary based static policy, noting that $\phi_i^{NT+1}(\mathcal{N})=\frac{N}{1+N\,(N+1)}$, we get $\mathsf{STA}=\sum_{i\in\mathcal{N}}\phi_i^{NT+1}(\mathcal{N})=\frac{N\,(N+1)}{1+N\,(N+1)}$, which is no larger than one. Thus, we have $\mathsf{STA}\leq 1$.

Noting that $\mathsf{OPT}\geq T+1-\frac{1}{N}$ for large enough N, we get $\frac{\mathsf{STA}}{\mathsf{OPT}}\leq\frac{1}{T+1-\frac{1}{N}}=\frac{1}{1+\|\mathcal{C}\|-\frac{1}{N}}$. By Theorem 5.1, we have $\frac{\mathsf{STA}}{\mathsf{OPT}}\geq\frac{1}{1+\|\mathcal{C}\|}$. Thus, $\frac{\mathsf{STA}}{\mathsf{OPT}}$ is arbitrarily close to $\frac{1}{1+\|\mathcal{C}\|}$ as N gets large.

Appendix E: Norm of a Collection of Intersection Preserving Subsets

In this section, we give a proof for Theorem 5.2. To see that the first statement holds, we construct a feasible solution to problem (10) that provides an objective value of $1+\lceil(\mathsf{D}_{\max}-1)/\mathsf{D}_{\min}\rceil$. In particular, we set $\hat{t}=1+\lceil(\mathsf{D}_{\max}-1)/\mathsf{D}_{\min}\rceil$. Furthermore, for each $[s,f]\in\mathcal{F}$, we set

$$\hat{C}_{[s,f]}=\left\{s+k\,\mathsf{D}_{\min}\;:\;k=0,1,2,\ldots,\left\lceil\frac{f-s}{\mathsf{D}_{\min}}\right\rceil-1\right\}\cup\{f\}. \tag{19}$$

The solution $\{\hat{C}_{[s,f]}:[s,f]\in\mathcal{F}\}$ and $\hat{t}$ provides an objective value of $\hat{t}=1+\lceil(\mathsf{D}_{\max}-1)/\mathsf{D}_{\min}\rceil$ for problem (10). We proceed to arguing that this solution is feasible for problem (10). Using the fact that $f-s+1\le\mathsf{D}_{\max}$ for all $[s,f]\in\mathcal{F}$, we have $\left|\hat{C}_{[s,f]}\right|\le\left\lceil\frac{f-s}{\mathsf{D}_{\min}}\right\rceil+1\le\left\lceil\frac{\mathsf{D}_{\max}-1}{\mathsf{D}_{\min}}\right\rceil+1=\hat{t}$. Thus, the first constraint is satisfied. The smallest element of $\hat{C}_{[s,f]}$ is s. Noting that $s+\left(\left\lceil\frac{f-s}{\mathsf{D}_{\min}}\right\rceil-1\right)\mathsf{D}_{\min}\le s+\frac{f-s}{\mathsf{D}_{\min}}\mathsf{D}_{\min}=f$, 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 $[s,f]\cap[a,b]\ne\varnothing$. We show that $\hat{C}_{[s,f]}\cap[a,b]\ne\varnothing$. 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 $\hat{C}_{[s,f]}\cap[a,b]\ne\varnothing$, as desired.

Second, consider the case $[s,f]\not\subseteq[a,b]$ and $[s,f]\not\supseteq[a,b]$. Because $[s,f]\cap[a,b]\ne\varnothing$, 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 $\hat{C}_{[s,f]}\cap[a,b]\ne\varnothing$, as desired.

Third, consider the case $[s,f]\supseteq[a,b]$. To get a contradiction, suppose, on the contrary, that $\hat{C}_{[s,f]}\cap[a,b]=\varnothing$. Since $[s,f]\supseteq[a,b]$, we have $s\le a\le b\le f$, but noting that $\hat{C}_{[s,f]}\cap[a,b]=\varnothing$ and $s,f\in\hat{C}_{[s,f]}$, there are two successive days c_1,c_2 in $\hat{C}_{[s,f]}$ such that $c_1<a\le b<c_2$. Thus, there are at least $b-a+1$ days in between days c_1 and c_2. Because $b-a+1\ge\mathsf{D}_{\min}$, there must be at least $\mathsf{D}_{\min}$ 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 $\mathsf{D}_{\min}-1$ 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 (10) as $|\,C_{[s,f]}\cap[a,b]\,|\;\ge\;\mathbb{1}_{\{[s,f]\cap[a,b]\ne\varnothing\}}$ for all $[s,f],[a,b]\in\mathcal{F}$. Define the constant $\mathsf{Z}^*_{[s,f]}$ as

$$\mathsf{Z}^*_{[s,f]}\;=\;\min\left\{\left|C_{[s,f]}\right|\;:\;\left|C_{[s,f]}\cap[a,b]\right|\;\ge\;\mathbb{1}_{\{[s,f]\cap[a,b]\ne\varnothing\}}\quad\forall\,[a,b]\in\mathcal{F},\quad C_{[s,f]}\subseteq[s,f]\right\}, \tag{20}$$

where the only decision variable is $C_{[s,f]}$. In this case, problem (10) becomes equivalent to $\min\{t:t\ge\mathsf{Z}^*_{[s,f]}\quad\forall\,[s,f]\in\mathcal{F}\}$, which has the optimal objective value $\max_{[s,f]\in\mathcal{F}}\mathsf{Z}^*_{[s,f]}$.

In the rest of the proof, we will show that we can compute $\mathsf{Z}^*_{[s,f]}$ in (20) by solving a minimization linear program with $O(\mathsf{D}_{\max})$ decision variables and $O(\mathsf{D}^2_{\max})$ constraints. Thus, letting χ^* be the optimal objective value of problem (10), by the discussion in the previous paragraph, we have $\chi^* = \max_{[s,f]\in\mathcal{F}} \mathsf{Z}^*_{[s,f]}$, which implies that χ^* is given by the maximum of the optimal objective values of $|\mathcal{F}| = O(\mathsf{D}_{\max} T)$ minimization linear programs, each of the linear programs having $O(\mathsf{D}_{\max})$ decision variables are $O(\mathsf{D}^2_{\max})$ 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 $O(|\mathcal{F}|\,\mathsf{D}_{\max}) = O(\mathsf{D}^2_{\max} T)$ decision variables and $O(|\mathcal{F}|\,\mathsf{D}^2_{\max}) = O(\mathsf{D}^3_{\max} T)$ constraints. To compute $\mathsf{Z}^*_{[s,f]}$ in (20) by solving a linear program, we use the decision variables $\{x_\ell : \ell = s,\ldots,f\} \in \{0,1\}^{f-s+1}$, 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

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

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 $\ell = s,\ldots,f$. Lastly, the right side of the constraint is nonzero for all $[a,b]\in\mathcal{F}$ such that $[s,f]\cap[a,b]\neq\varnothing$ and there are only $O(\mathsf{D}^2_{\max})$ such constraints. Thus, the problem above actually has $f-s+1 = O(\mathsf{D}_{\max})$ decision variables and $O(\mathsf{D}^2_{\max})$ constraints.

Appendix F: Examples of Intersection Preserving Subsets

By the discussion in Section 5, setting $C_{[s,f]} = \{s, s+1,\ldots,f\}$ trivially yields an intersection preserving subset for the interval $[s,f]$. In this case, we immediately obtain a collection of intersection preserving subsets $\{\{s,s+1,\ldots,f\} : [s,f]\in\mathcal{F}\}$. The norm of this collection is $\max_{[s,f]\in\mathcal{F}}\{f-s+1\} = \mathsf{D}_{\max}$. Also, by Theorem 5.2, we always have a collection of intersection preserving subsets with norm at most $1+\lceil(\mathsf{D}_{\max}-1)/\mathsf{D}_{\min}\rceil$. These results hold for any arbitrary set of intervals $\mathcal{F}$ for which we may get booking requests. In this section, we give examples of intersection preserving subsets when there is a special structure in the set of intervals for which we may get booking requests. We may end up with collections of intersection preserving subsets with smaller norms when there is a special structure in the set of intervals. Smaller norms for the collection of intersection preserving subsets translate into better performance guarantees for the itinerary based static policy. We consider two special settings.

First, consider the setting where there exists a set of days $\{\tau_1, \ldots, \tau_K\}$ such that each booking request starts on one of these days. In this case, we argue that setting $C_{[s,f]} = [s,f] \cap \{\tau_1, \ldots, \tau_K\}$ yields an intersection preserving subset for the interval $[s,f]$. With this definition of an intersection preserving subset, we clearly have $C_{[s,f]} \subseteq [s,f]$, so the first property of an intersection preserving subset is satisfied. To check that the second property is satisfied, let the interval $[a,b] \in \mathcal{F}$ be such that $[s,f] \cap [a,b] \neq \varnothing$. Consider the case $s \le a$. Because $[s,f] \cap [a,b] \neq \varnothing$, having $s \le a$ implies that $f \ge a$, yielding $s \le a \le f$. Thus, we have $a \in [s,f]$. Noting that all booking requests start on one of the days $\{\tau_1, \ldots, \tau_K\}$, we have $a \in \{\tau_1, \ldots, \tau_K\}$. In this case, we obtain $a \in [s,f] \cap \{\tau_1, \ldots, \tau_K\}$, so $a \in C_{[s,f]}$. Thus, we have $C_{[s,f]} \cap [a,b] \neq \varnothing$, so the second property of an intersection preserving subset is satisfied. We can use the same approach to show that $C_{[s,f]} \cap [a,b] \neq \varnothing$ when $s \ge a$.

Second, consider the setting where each booking request is for the same number of days, so $s - f = L$ for all $[s,f] \in \mathcal{F}$. In this case, we argue that setting $C_{[s,f]} = \{s,f\}$ yields an intersection preserving subset for the interval $[s,f]$. With this definition of an intersection preserving subset, we clearly have $C_{[s,f]} = \{s,f\} \subseteq [s,f]$, so the first property of an intersection preserving subset is satisfied. To check that the second property is satisfied, assume on the contrary that there exists an interval $[a,b]$ such that $[s,f] \cap [a,b] \neq \varnothing$ and $C_{[s,f]} \cap [a,b] = \varnothing$. Because $C_{[s,f]} = \{s,f\}$, having $C_{[s,f]} \cap [a,b] = \varnothing$ implies that $s \notin [a,b]$ and $f \notin [a,b]$, in which case, having $[s,f] \cap [a,b] \neq \varnothing$ implies that $[a,b] \subseteq [s+1, f-1]$. If $[a,b] \subseteq [s+1, f-1]$, then the booking requests for the interval $[a,b]$ are at most for $f - s - 1$ days, but the booking requests for the interval $[s,f]$ are for $f - s + 1$ days, contradicting the fact that each booking request is for the same number of days.

Appendix G: Upper Bound on the Optimal Policy Performance

In this section, we give the proofs of the three results in Section 6, along with Lemma G.1 that we use in that section. Here is the proof of Proposition 6.1.

Proof of Proposition 6.1:

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

$$\sum_{j \in \mathcal{N}} \varphi_j \sum_{i \in \mathcal{N} \setminus \{j\}} \beta^q_{j,[s,f] \to i} \;=\; \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N} \setminus \{i\}} \varphi_j \, \beta^q_{j,[s,f] \to i} \;=\; \sum_{j \in \mathcal{N}} \sum_{i \in \mathcal{N} \setminus \{j\}} \varphi_i \, \beta^q_{i,[s,f] \to j}, \tag{21}$$

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 $J^{Q+1} = 0 = \sum_{j\in\mathcal{N}} V^{Q+1}_{\boldsymbol{\beta},j}$, 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 $\boldsymbol{x} \in \{0,1\}^{N\times T}$ 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, $J^{q+1}(\boldsymbol{x} - \boldsymbol{e}_{j,[s,f]}) \le \sum_{i\in\mathcal{N}\setminus\{j\}} V^{q+1}_{\boldsymbol{\beta},i}(\boldsymbol{x}_i) + V^{q+1}_{\boldsymbol{\beta},j}(\boldsymbol{x}_j - \boldsymbol{e}_{[s,f]})$ and $J^{q+1}(\boldsymbol{x}) \le \sum_{j\in\mathcal{N}} V^{q+1}_{\boldsymbol{\beta},j}(\boldsymbol{x}_j)$. Using (1), we get

$$
\begin{aligned}
J^q(\boldsymbol{x}) \;&=\; \sum_{[s,f]\in\mathcal{F}} \lambda^q_{[s,f]} \sum_{j\in\mathcal{N}} \phi^q_j(S^*_{[s,f]}) \left(\prod_{\ell=s}^{f} x_{j,\ell}\right) \Big[r_{j,[s,f]} + J^{q+1}(\boldsymbol{x} - \boldsymbol{e}_{j,[s,f]})) - J^{q+1}(\boldsymbol{x}) \Big] + J^{q+1}(\boldsymbol{x}) \\
&\overset{(a)}{\le} \sum_{[s,f]\in\mathcal{F}} \lambda^q_{[s,f]} \sum_{j\in\mathcal{N}} \phi^q_j(S^*_{[s,f]}) \left(\prod_{\ell=s}^{f} x_{j,\ell}\right) \Big[r_{j,[s,f]} + V^{q+1}_{\boldsymbol{\beta},j}(\boldsymbol{x}_j - \boldsymbol{e}_{[s,f]})) - V^{q+1}_{\boldsymbol{\beta},j}(\boldsymbol{x}_j) \Big] + \sum_{j\in\mathcal{N}} V^{q+1}_{\boldsymbol{\beta},j}(\boldsymbol{x}_j) \\
&\overset{(b)}{=} \sum_{[s,f]\in\mathcal{F}} \lambda^q_{[s,f]} \Bigg\{ \sum_{j\in\mathcal{N}} \phi^q_j(S^*_{[s,f]}) \left(\prod_{\ell=s}^{f} x_{j,\ell}\right) \Big[\beta^q_{j,[s,f]\to j} + V^{q+1}_{\boldsymbol{\beta},j}(\boldsymbol{x}_j - \boldsymbol{e}_{[s,f]}) - V^{q+1}_{\boldsymbol{\beta},j}(\boldsymbol{x}_j) \Big] \\
&\qquad\qquad + \sum_{j\in\mathcal{N}} \sum_{i\in\mathcal{N}\setminus\{j\}} \phi^q_i(S^*_{[s,f]}) \left(\prod_{\ell=s}^{f} x_{i,\ell}\right) \beta^q_{i,[s,f]\to j} \Bigg\} + \sum_{j\in\mathcal{N}} V^{q+1}_{\boldsymbol{\beta},j}(\boldsymbol{x}_j) \\
&\overset{(c)}{=} \sum_{j\in\mathcal{N}} \Bigg\{ \sum_{[s,f]\in\mathcal{F}} \lambda^q_{[s,f]} \Bigg\{ \phi^q_j(S^*_{[s,f]}) \left(\prod_{\ell=s}^{f} x_{j,\ell}\right) \Big[\beta^q_{j,[s,f]\to j} + V^{q+1}_{\boldsymbol{\beta},j}(\boldsymbol{x}_j - \boldsymbol{e}_{[s,f]}) - V^{q+1}_{\boldsymbol{\beta},j}(\boldsymbol{x}_j) \Big] \\
&\qquad\qquad + \sum_{i\in\mathcal{N}\setminus\{j\}} \phi^q_i(S^*_{[s,f]}) \left(\prod_{\ell=s}^{f} x_{i,\ell}\right) \beta^q_{i,[s,f]\to j} \Bigg\} + V^{q+1}_{\boldsymbol{\beta},j}(\boldsymbol{x}_j) \Bigg\} \\
&\le \sum_{j\in\mathcal{N}} \Bigg\{ \sum_{[s,f]\in\mathcal{F}} \lambda^q_{[s,f]} \max_{S\subseteq\mathcal{N}} \Bigg\{ \phi^q_j(S) \left(\prod_{\ell=s}^{f} x_{j,\ell}\right) \Big[\beta^q_{j,[s,f]\to j} + V^{q+1}_{\boldsymbol{\beta},j}(\boldsymbol{x}_j - \boldsymbol{e}_{[s,f]}) - V^{q+1}_{\boldsymbol{\beta},j}(\boldsymbol{x}_j) \Big] \\
&\qquad\qquad + \sum_{i\in\mathcal{N}\setminus\{j\}} \phi^q_i(S) \left(\prod_{\ell=s}^{f} x_{i,\ell}\right) \beta^q_{i,[s,f]\to j} \Bigg\} + V^{q+1}_{\boldsymbol{\beta},j}(\boldsymbol{x}_j) \Bigg\} \\
&\overset{(d)}{\le} \sum_{j\in\mathcal{N}} \Bigg\{ \sum_{[s,f]\in\mathcal{F}} \lambda^q_{[s,f]} \max_{S\subseteq\mathcal{N}} \Bigg\{ \phi^q_j(S) \left(\prod_{\ell=s}^{f} x_{j,\ell}\right) \Big[\beta^q_{j,[s,f]\to j} + V^{q+1}_{\boldsymbol{\beta},j}(\boldsymbol{x}_j - \boldsymbol{e}_{[s,f]}) - V^{q+1}_{\boldsymbol{\beta},j}(\boldsymbol{x}_j) \Big] \\
&\qquad\qquad + \sum_{i\in\mathcal{N}\setminus\{j\}} \phi^q_i(S)\, \beta^q_{i,[s,f]\to j} \Bigg\} + V^{q+1}_{\boldsymbol{\beta},j}(\boldsymbol{x}_j) \Bigg\} \\
&\overset{(e)}{=} \sum_{j\in\mathcal{N}} V^q_{\boldsymbol{\beta},j}(\boldsymbol{x}_j).
\end{aligned}
$$

In the chain of inequalities above, (a) uses the induction hypothesis. To see that (b) holds, we note that $r_{j,[s,f]} = \sum_{i\in\mathcal{N}} \beta^q_{j,[s,f]\to i} = \beta^q_{j,[s,f]\to j} + \sum_{i\in\mathcal{N}\setminus\{j\}} \beta^q_{j,[s,f]\to i}$ and use (21) after identifying φ_j with $\phi^q_j(S^*_{[s,f]}) \prod_{\ell=s}^{f} x_{j,\ell}$. 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 B.4. Lastly, (e) follows from (12). The chain of inequalities above completes the induction argument. ■

Next, we state and prove Lemma G.1.

Lemma G.1 (Equivalence of Linear Programs) *Problems* (11) *and* (13) *have the same optimal objective value.*

Proof: We let $\hat{\boldsymbol{h}} = \{\hat{h}^q_{[s,f]}(S) : S \subseteq \mathcal{N},\ [s,f] \in \mathcal{F},\ q \in \mathcal{Q}\}$ and $\hat{\boldsymbol{y}} = \{\hat{y}_{i,[s,f]} : i \in \mathcal{N},\ [s,f] \in \mathcal{F}\}$ be an optimal solution to problem (11). For each $j \in \mathcal{N}$, we set $\widetilde{h}^q_{[s,f]\to j}(S) = \hat{h}^q_{[s,f]}(S)$. In this case, we observe that the solution $\widetilde{\boldsymbol{h}} = \{\widetilde{h}^q_{[s,f]\to j}(S) : j \in \mathcal{N},\ S \subseteq \mathcal{N},\ [s,f] \in \mathcal{F},\ q \in \mathcal{Q}\}$ and $\hat{\boldsymbol{y}}$ is feasible to problem (13) and provides the same objective value as the optimal objective value of problem (11). Therefore, the optimal objective value of problem (13) is at least as large as that of problem (11). In the rest of the proof, we show that the reverse inequality also holds. We let $\widetilde{\boldsymbol{h}} = \{\widetilde{h}^q_{[s,f]\to j}(S) : j \in \mathcal{N},\ S \subseteq \mathcal{N},\ [s,f] \in \mathcal{F},\ q \in \mathcal{Q}\}$ and $\widetilde{\boldsymbol{y}} = \{\widetilde{y}_{i,[s,f]} : i \in \mathcal{N},\ [s,f] \in \mathcal{F}\}$ be an optimal solution to problem (13). We define $\hat{h}^q_{[s,f]}(S)$ as

$$\hat{h}^q_{[s,f]}(S) = \frac{1}{N} \sum_{j \in \mathcal{N}} \widetilde{h}^q_{[s,f]\to j}(S).$$

We establish that the solution $\hat{\boldsymbol{h}} = \left\{\hat{h}^q_{[s,f]}(S) : S \subseteq \mathcal{N},\ [s,f] \in \mathcal{F},\ q \in \mathcal{Q}\right\}$ and $\widetilde{\boldsymbol{y}}$ is feasible to problem (11). Using the definition of $\hat{h}^q_{[s,f]}(S)$ above, we have

$$\sum_{S \subseteq \mathcal{N}} \hat{h}^q_{[s,f]}(S) \;=\; \frac{1}{N} \sum_{j \in \mathcal{N}} \sum_{S \subseteq \mathcal{N}} \widetilde{h}^q_{[s,f]\to j}(S) \;\overset{(a)}{=}\; \frac{1}{N} \sum_{j \in \mathcal{N}} \lambda^q_{[s,f]} \;=\; \lambda^q_{[s,f]},$$

where (a) holds because the solution $(\widetilde{\boldsymbol{h}}, \widetilde{\boldsymbol{y}})$ satisfies the second constraint in problem (13). Thus, the solution $(\hat{\boldsymbol{h}}, \widetilde{\boldsymbol{y}})$ satisfies the second constraint in problem (11).

To check that the solution $(\hat{\boldsymbol{h}}, \widetilde{\boldsymbol{y}})$ satisfies the third constraint in problem (11), using the definition of $\hat{h}^q_{[s,f]}(S)$ once more, we have

$$\sum_{S \subseteq \mathcal{N}} \phi^q_i(S)\, \hat{h}^q_{[s,f]}(S) \;=\; \frac{1}{N} \sum_{j \in \mathcal{N}} \sum_{S \subseteq \mathcal{N}} \phi^q_i(S)\, \widetilde{h}^q_{[s,f]\to j}(S) \;\overset{(b)}{=}\; \frac{1}{N} \sum_{j \in \mathcal{N}} \widetilde{y}^q_{i,[s,f]} \;=\; \widetilde{y}^q_{i,[s,f]}, \tag{22}$$

where (b) holds because the solution $(\widetilde{\boldsymbol{h}}, \widetilde{\boldsymbol{y}})$ satisfies the third constraint in problem (13). Thus, the solution $(\hat{\boldsymbol{h}}, \widetilde{\boldsymbol{y}})$ satisfies the third constraint in problem (11).

Lastly, we check that the solution $(\hat{\boldsymbol{h}}, \widetilde{\boldsymbol{y}})$ satisfies the first constraint in problem (11). In particular, we have the chain of inequalities

$$\begin{aligned}\sum_{q \in \mathcal{Q}} \sum_{[s,f] \in \mathcal{F}} \sum_{S \subseteq \mathcal{N}} \mathbb{1}_{\{\ell \in [s,f]\}}\ \phi^q_i(S)\ \hat{h}^q_{[s,f]}(S) \;&=\; \sum_{q \in \mathcal{Q}} \sum_{[s,f] \in \mathcal{F}} \mathbb{1}_{\{\ell \in [s,f]\}} \sum_{S \subseteq \mathcal{N}} \phi^q_i(S)\ \hat{h}^q_{[s,f]}(S) \\ &\overset{(c)}{=}\; \sum_{q \in \mathcal{Q}} \sum_{[s,f] \in \mathcal{F}} \mathbb{1}_{\{\ell \in [s,f]\}}\, \widetilde{y}^q_{i,[s,f]} \;\overset{(d)}{=}\; \sum_{q \in \mathcal{Q}} \sum_{[s,f] \in \mathcal{F}} \mathbb{1}_{\{\ell \in [s,f]\}} \sum_{S \subseteq \mathcal{N}} \phi^q_i(S)\, \widetilde{h}_{[s,f]\to i}(S) \;\overset{(e)}{\leq}\; 1,\end{aligned}$$

where (c) follows from (22), (d) follows from the fact that the solution $(\widetilde{\boldsymbol{h}}, \widetilde{\boldsymbol{y}})$ satisfies the third constraint in problem (13) with $i = j$, and (e) holds because the solution $(\widetilde{\boldsymbol{h}}, \widetilde{\boldsymbol{y}})$ also satisfies the first constraint in problem (13). Therefore, the solution $(\hat{\boldsymbol{h}}, \widetilde{\boldsymbol{y}})$ satisfies the first constraint in problem (11). Thus, the solution $(\hat{\boldsymbol{h}}, \widetilde{\boldsymbol{y}})$ is feasible to problem (11). The solution $(\hat{\boldsymbol{h}}, \widetilde{\boldsymbol{y}})$ provides the objective value $\sum_{q\in\mathcal{Q}} \;\sum_{i\in\mathcal{N}}\sum_{[s,f]\in\mathcal{F}} r_{i,[s,f]}\, \widetilde{y}^q_{i,[s,f]}$, which is the optimal objective value of problem (13). So, the optimal objective value of problem (11) is at least as large as that of problem (13). ∎

To write the dual of problem (13), associating the dual variables $\boldsymbol{\mu} = \{\mu_{i,\ell} : i\in\mathcal{N},\ \ell\in\mathcal{T}\}$, $\boldsymbol{\sigma} = \{\sigma^q_{[s,f]\to j} : j\in\mathcal{N},\ [s,f]\in\mathcal{F},\ q\in\mathcal{Q}\}$ and $\boldsymbol{\beta} = \{\beta^q_{i,[s,f]\to j} : i,j\in\mathcal{N},\ [s,f]\in\mathcal{F},\ q\in\mathcal{Q}\}$, we have

$$
\begin{aligned}
\min \quad & \sum_{i\in\mathcal{N}}\sum_{\ell\in\mathcal{T}} \mu_{i,\ell} \;+\; \sum_{q\in\mathcal{Q}}\sum_{j\in\mathcal{N}}\sum_{[s,f]\in\mathcal{F}} \lambda^q_{[s,f]}\,\sigma^q_{[s,f]\to j} && (23)\\
\text{st} \quad & \sigma^q_{[s,f]\to j} \;\geq\; \sum_{i\in\mathcal{N}} \phi^q_i(S)\,\beta^q_{i,[s,f]\to j} \;-\; \phi^q_j(S)\sum_{\ell=s}^{f}\mu_{j,\ell} \quad \forall\, j\in\mathcal{N},\ [s,f]\in\mathcal{F},\ S\subseteq\mathcal{N},\ q\in\mathcal{Q}\\
& \sum_{j\in\mathcal{N}} \beta^q_{i,[s,f]\to j} \;=\; r_{i,[s,f]} \quad \forall\, i\in\mathcal{N},\ [s,f]\in\mathcal{F}\\
& \boldsymbol{\mu}\geq \mathbf{0},\ \boldsymbol{\sigma},\boldsymbol{\beta} \text{ free.}
\end{aligned}
$$

It is simple to check that problem (13) is feasible and bounded, so by strong duality, problem (23) also has the optimal objective value $\mathsf{Z}^*_{\mathsf{LP}}$. We use problem (23) to give a proof for Theorem 6.2.

Proof of Theorem 6.2:

Let $(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\sigma}}, \hat{\boldsymbol{\beta}})$ be an optimal solution to problem (23). For notational brevity, also letting $\hat{\alpha}^q_j = \sum_{k=q}^{Q}\sum_{[s,f]\in\mathcal{F}} \lambda^k_{[s,f]}\,\hat{\sigma}^k_{[s,f]\to j}$, we will use induction over the time periods to establish that $V^q_{\hat{\boldsymbol{\beta}},j}(\boldsymbol{x}_j) \leq \hat{\alpha}^q_j + \sum_{\ell\in\mathcal{T}} \hat{\mu}_{j,\ell}\, x_{j,\ell}$ for all $\boldsymbol{x}_j\in\{0,1\}^T$, $j\in\mathcal{N}$ and $q\in\mathcal{Q}$. In this case, using this result with $q=1$ and $\boldsymbol{x}_j = \boldsymbol{e}'$, we obtain $\sum_{j\in\mathcal{N}} V^1_{\hat{\boldsymbol{\beta}},j}(\boldsymbol{e}') \leq \sum_{j\in\mathcal{N}} \hat{\alpha}^1_j + \sum_{j\in\mathcal{N}}\sum_{\ell\in\mathcal{T}} \hat{\mu}_{j,\ell} = \mathsf{Z}^*_{\mathsf{LP}}$, where the equality uses the fact that $\sum_{j\in\mathcal{N}} \hat{\alpha}^1_j + \sum_{j\in\mathcal{N}}\sum_{\ell\in\mathcal{T}}\hat{\mu}_{j,\ell} = \sum_{j\in\mathcal{N}}\sum_{q\in\mathcal{Q}}\sum_{[s,f]\in\mathcal{F}} \lambda^q_{[s,f]}\,\hat{\sigma}^q_{[s,f]\to j} + \sum_{j\in\mathcal{N}}\sum_{\ell\in\mathcal{T}}\hat{\mu}_{j,\ell}$ and the last quantity is the optimal objective value of problem (23), which is $\mathsf{Z}^*_{\mathsf{LP}}$. Thus, the desired result follows. In the rest of the proof, we focus on the induction argument. Noting that $\hat{\boldsymbol{\mu}}\geq 0$, for each $\boldsymbol{x}_j\in\{0,1\}^T$, at time period $Q+1$, we have $V^{Q+1}_{\hat{\boldsymbol{\beta}},j}(\boldsymbol{x}_j) = 0 \leq \sum_{\ell\in\mathcal{T}}\hat{\mu}_{j,\ell}\,x_{j,\ell}$, 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 $V^{q+1}_{\hat{\boldsymbol{\beta}},j}(\boldsymbol{x}_j)$ and $V^{q+1}_{\hat{\boldsymbol{\beta}},j}(\boldsymbol{x}_j - \boldsymbol{e}_{[s,f]})$ are upper bounded by $\hat{\alpha}^{q+1}_j + \sum_{\ell\in\mathcal{T}}\hat{\mu}_{j,\ell}\,x_{j,\ell}$ and $\hat{\alpha}^{q+1}_j + \sum_{\ell\in\mathcal{T}}\hat{\mu}_{j,\ell}\,x_{j,\ell} - \sum_{\ell=s}^{f}\hat{\mu}_{j,\ell}$. So, by (12),

$$
\begin{aligned}
V^q_{\hat{\boldsymbol{\beta}},j}(\boldsymbol{x}_j) \leq \sum_{[s,f]\in\mathcal{F}} \lambda^q_{[s,f]}\, \max_{S\subseteq\mathcal{N}} \left\{ \phi^q_j(S)\left(\prod_{\ell=s}^{f} x_{j,\ell}\right)\left[\beta^q_{j,[s,f]\to j} - \sum_{\ell=s}^{f}\hat{\mu}_{j,\ell}\right] + \sum_{i\in\mathcal{N}\setminus\{j\}} \phi^q_i(S)\;\beta^q_{i,[s,f]\to j}\right\}&\\
+\,\hat{\alpha}^{q+1}_j + \sum_{\ell\in\mathcal{T}}\hat{\mu}_{j,\ell}\,x_{j,\ell}.&
\end{aligned}
$$

Following the same argument that we used to obtain the inequality (a) in the proof of Proposition B.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 $S^*_{[s,f]\to j}$ be an optimal solution to the resulting maximization problem. In this case, using the inequality above, we get

$$\begin{aligned}
V^q_{\hat{\boldsymbol{\beta}},j}(\boldsymbol{x}_j) \;\le\; & \sum_{[s,f]\in\mathcal{F}} \lambda^q_{[s,f]} \max_{S\subseteq\mathcal{N}} \left\{ \phi^q_j(S)\left[\beta^q_{j,[s,f]\to j} - \sum_{\ell=s}^{f}\hat{\mu}_{j,\ell}\right] + \sum_{i\in\mathcal{N}\setminus\{j\}} \phi^q_i(S)\,\beta^q_{i,[s,f]\to j}\right\} \\
& \qquad\qquad + \hat{\alpha}^{q+1}_j + \sum_{\ell\in\mathcal{T}}\hat{\mu}_{j,\ell}\,x_{j,\ell} \\
\overset{(a)}{=} & \sum_{[s,f]\in\mathcal{F}} \lambda^q_{[s,f]} \left\{ \phi^q_j(S^*_{[s,f]\to j})\left[\beta^q_{j,[s,f]\to j} - \sum_{\ell=s}^{f}\hat{\mu}_{j,\ell}\right] + \sum_{i\in\mathcal{N}\setminus\{j\}} \phi^q_i(S^*_{[s,f]\to j})\,\beta^q_{i,[s,f]\to j}\right\} + \hat{\alpha}^{q+1}_j + \sum_{\ell\in\mathcal{T}}\hat{\mu}_{j,\ell}\,x_{j,\ell} \\
\overset{(b)}{=} & \sum_{[s,f]\in\mathcal{F}} \lambda^q_{[s,f]} \left\{ \sum_{i\in\mathcal{N}} \phi^q_i(S^*_{[s,f]\to j})\,\beta^q_{i,[s,f]\to j} - \phi^q_j(S^*_{[s,f]\to j}) \sum_{\ell=s}^{f}\hat{\mu}_{j,\ell}\right\} + \hat{\alpha}^{q+1}_j + \sum_{\ell\in\mathcal{T}}\hat{\mu}_{j,\ell}\,x_{j,\ell} \\
\overset{(c)}{\le} & \sum_{[s,f]\in\mathcal{F}} \lambda^q_{[s,f]}\,\hat{\sigma}^q_{[s,f]\to j} + \hat{\alpha}^{q+1}_j + \sum_{\ell\in\mathcal{T}}\hat{\mu}_{j,\ell}\,x_{j,\ell} \;\overset{(d)}{=}\; \hat{\alpha}^q_j + \sum_{\ell\in\mathcal{T}}\hat{\mu}_{j,\ell}\,x_{j,\ell},
\end{aligned}$$

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

We focus on the proof of Theorem 6.3. To give an alternative representation of the value functions $\{\Gamma^q_{\boldsymbol{\beta},j} : q\in\mathcal{Q}\}$ in (14), for each $\ell\in\mathcal{T}$, define the value function

$$\Psi^q_{\boldsymbol{\beta},j}(\ell) \;=\; \sum_{k=q}^{Q}\;\sum_{[s,f]\in\mathcal{F}} \mathbb{1}_{\{s=\ell\}}\,\lambda^k_{[s,f]} \max_{S\subseteq\mathcal{N}\setminus\{j\}} \left\{ \sum_{i\in\mathcal{N}\setminus\{j\}} \phi^k_i(S)\,\beta^k_{i,[s,f]\to j}\right\}. \tag{24}$$

Directly by comparing (24) with (14), observe that the value functions $\{\Gamma^q_{\boldsymbol{\beta},j} : q\in\mathcal{Q}\}$ and $\{\Psi^q_{\boldsymbol{\beta},j} : q\in\mathcal{Q}\}$ satisfy the relationship $\Gamma^q_{\boldsymbol{\beta},j}(a,b) = \sum_{\ell=a}^{b}\Psi^q_{\boldsymbol{\beta},j}(\ell)$ for each interval $[a,b]$. Given that the state of resource j is $\boldsymbol{x}_j\in\{0,1\}^T$, let $\mathcal{K}(\boldsymbol{x}_j)$ be the set of unavailable days for this resource; that is $\ell\in\mathcal{K}(\boldsymbol{x}_j)$ if and only if $x_{j,\ell}=0$. Note that $\mathcal{K}(\boldsymbol{x}_j)$ is the union of the maximal unavailable intervals with respect to $\boldsymbol{x}_j$. In other words, we have $\mathcal{K}(\boldsymbol{x}_j) = \cup_{[a,b]\in\mathcal{H}(\boldsymbol{x}_j)}[a,b]$. In this case, by the discussion earlier in this paragraph, we obtain the identity

$$\sum_{[a,b]\in\mathcal{H}(\boldsymbol{x}_j)} \Gamma^q_{\boldsymbol{\beta},j}(a,b) \;=\; \sum_{[a,b]\in\mathcal{H}(\boldsymbol{x}_j)}\;\sum_{\ell\in[a,b]} \Psi^q_{\boldsymbol{\beta},j}(\ell) \;=\; \sum_{\ell\in\mathcal{K}(\boldsymbol{x}_j)} \Psi^q_{\boldsymbol{\beta},j}(\ell),$$

where the last equality holds since $\mathcal{K}(\boldsymbol{x}_j) = \cup_{[a,b]\in\mathcal{H}(\boldsymbol{x}_j)}[a,b]$. Thus, to establish Theorem 6.3, it is enough to show that $V^q_{\boldsymbol{\beta},j}(\boldsymbol{x}_j) = \sum_{[a,b]\in\mathcal{I}(\boldsymbol{x}_j)}\Theta^q_{\boldsymbol{\beta},j}(a,b) + \sum_{\ell\in\mathcal{K}(\boldsymbol{x}_j)}\Psi^q_{\boldsymbol{\beta},j}(\ell)$.

Proof of Theorem 6.3:

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

$Q+1$, we have $V^{Q+1}_{\boldsymbol{\beta},j} = \Theta^{Q+1}_{\boldsymbol{\beta},j} = \Psi^{Q+1}_{\boldsymbol{\beta},j} = 0$, 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 $\boldsymbol{x}_j \in \{0,1\}^T$, the collection of maximal available intervals $\mathcal{I}(\boldsymbol{x}_j)$ and the set of unavailable days $\mathcal{K}(\boldsymbol{x}_j)$ collectively cover $\mathcal{T}$; that is, $\mathcal{T} = \big(\cup_{[a,b]\in\mathcal{I}(\boldsymbol{x}_j)}[a,b]\big) \cup \mathcal{K}(\boldsymbol{x}_j)$. Thus, for each $s \in \mathcal{T}$, we have $\sum_{[a,b]\in\mathcal{I}(\boldsymbol{x}_j)} \mathbb{1}_{\{s\in[a,b]\}} + \sum_{\ell\in\mathcal{K}(\boldsymbol{x}_j)} \mathbb{1}_{\{s=\ell\}} = 1$. In this case, by (12), we get

$$
\begin{aligned}
V^q_{\boldsymbol{\beta},j}(\boldsymbol{x}_j) \;=\; & V^{q+1}_{\boldsymbol{\beta},j}(\boldsymbol{x}_j) + \sum_{[s,f]\in\mathcal{F}} \lambda^q_{[s,f]} \left(\sum_{[a,b]\in\mathcal{I}(\boldsymbol{x}_j)} \mathbb{1}_{\{s\in[a,b]\}} + \sum_{\ell\in\mathcal{K}(\boldsymbol{x}_j)} \mathbb{1}_{\{s=\ell\}} \right) \times \\
& \max_{S\subseteq\mathcal{N}} \left\{ \phi^q_j(S) \left(\prod_{h=s}^{f} x_{j,h} \right) \left[\beta^q_{j,[s,f]\to j} + V^{q+1}_{\boldsymbol{\beta},j}(\boldsymbol{x}_j - \boldsymbol{e}_{[s,f]}) - V^{q+1}_{\boldsymbol{\beta},j}(\boldsymbol{x}_j) \right] + \sum_{i\in\mathcal{N}\setminus\{j\}} \phi^q_i(S)\, \beta^q_{i,[s,f]\to j} \right\} \\
= & \sum_{[a,b]\in\mathcal{I}(\boldsymbol{x}_j)} \Bigg(\Theta^{q+1}_{\boldsymbol{\beta},j}(a,b) \;+\; \sum_{[s,f]\in\mathcal{F}} \mathbb{1}_{\{s\in[a,b]\}}\, \lambda^q_{[s,f]} \times \\
& \max_{S\subseteq\mathcal{N}} \left\{ \phi^q_j(S) \left(\prod_{h=s}^{f} x_{j,h} \right) \left[\beta^q_{j,[s,f]\to j} + V^{q+1}_{\boldsymbol{\beta},j}(\boldsymbol{x}_j - \boldsymbol{e}_{[s,f]}) - V^{q+1}_{\boldsymbol{\beta},j}(\boldsymbol{x}_j) \right] + \sum_{i\in\mathcal{N}\setminus\{j\}} \phi^q_i(S)\, \beta^q_{i,[s,f]\to j} \right\} \Bigg) \\
& + \sum_{\ell\in\mathcal{K}(\boldsymbol{x}_j)} \Bigg(\Psi^{q+1}_{\boldsymbol{\beta},j}(\ell) \;+\; \sum_{[s,f]\in\mathcal{F}} \mathbb{1}_{\{s=\ell\}}\, \lambda^q_{[s,f]} \times \\
& \max_{S\subseteq\mathcal{N}} \left\{ \phi^q_j(S) \left(\prod_{h=s}^{f} x_{j,h} \right) \left[\beta^q_{j,[s,f]\to j} + V^{q+1}_{\boldsymbol{\beta},j}(\boldsymbol{x}_j - \boldsymbol{e}_{[s,f]}) - V^{q+1}_{\boldsymbol{\beta},j}(\boldsymbol{x}_j) \right] + \sum_{i\in\mathcal{N}\setminus\{j\}} \phi^q_i(S)\, \beta^q_{i,[s,f]\to j} \right\} \Bigg),
\end{aligned}
$$

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

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

$$
\begin{aligned}
\Theta^q_{\boldsymbol{\beta},j}(a,b) \;=\; & \Theta^{q+1}_{\boldsymbol{\beta},j}([a,b]) + \sum_{[s,f]\in\mathcal{F}} \mathbb{1}_{\{s\in[a,b]\}}\, \lambda^q_{[s,f]} \times \\
& \max_{S\subseteq\mathcal{N}} \left\{ \phi^q_j(S) \left(\prod_{h=s}^{f} x_{j,h} \right) \left[\beta^q_{j,[s,f]\to j} + V^{q+1}_{\boldsymbol{\beta},j}(\boldsymbol{x}_j - \boldsymbol{e}_{[s,f]}) - V^{q+1}_{\boldsymbol{\beta},j}(\boldsymbol{x}_j) \right] + \sum_{i\in\mathcal{N}\setminus\{j\}} \phi^q_i(S)\, \beta^q_{i,[s,f]\to j} \right\}, \quad (25) \\
\Psi^q_{\boldsymbol{\beta},j}(\ell) \;=\; & \Psi^{q+1}_{\boldsymbol{\beta},j}(\ell) + \sum_{[s,f]\in\mathcal{F}} \mathbb{1}_{\{s=\ell\}}\, \lambda^q_{[s,f]} \times \\
& \max_{S\subseteq\mathcal{N}} \left\{ \phi^q_j(S) \left(\prod_{h=s}^{f} x_{j,h} \right) \left[\beta^q_{j,[s,f]\to j} + V^{q+1}_{\boldsymbol{\beta},j}(\boldsymbol{x}_j - \boldsymbol{e}_{[s,f]}) - V^{q+1}_{\boldsymbol{\beta},j}(\boldsymbol{x}_j) \right] + \sum_{i\in\mathcal{N}\setminus\{j\}} \phi^q_i(S)\, \beta^q_{i,[s,f]\to j} \right\}. \quad (26)
\end{aligned}
$$

Thus, by the first displayed equality in the proof, $V^q_{\boldsymbol{\beta},j}(\boldsymbol{x}_j) = \sum_{[a,b]\in\mathcal{I}(\boldsymbol{x}_j)} \Theta^q_{\boldsymbol{\beta},j}(a,b) + \sum_{\ell\in\mathcal{K}(\boldsymbol{x}_j)} \Psi^q_{\boldsymbol{\beta},j}(\ell)$, completing the induction argument.

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

$s \in [a,b]$ and $f \in [a,b]$, then because $[a,b] \in \mathcal{I}(\boldsymbol{x}_j)$, we get $x_{j,h} = 1$ for all $h = s,\ldots,f$, 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 $h = s+1,\ldots,f$ 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 $\mathbb{1}_{\{s\in[a,b]\}} = \mathbb{1}_{\{s\in[a,b],\ [s,f]\subseteq[a,b]\}} + \mathbb{1}_{\{s\in[a,b],\ [s,f]\not\subseteq[a,b]\}}$, we equivalently express the right side of (25) as

$$
\begin{aligned}
&\Theta^{q+1}_{\boldsymbol{\beta},j}(a,b) + \sum_{[s,f]\in\mathcal{F}} \mathbb{1}_{\{s\in[a,b],\ [s,f]\subseteq[a,b]\}}\, \lambda^q_{[s,f]} \times \\
&\qquad \max_{S\subseteq\mathcal{N}} \left\{ \phi^q_j(S) \left(\prod_{h=s}^{f} x_{j,h} \right) \left[\beta^q_{j,[s,f]\to j} + V^{q+1}_{\boldsymbol{\beta},j}(\boldsymbol{x}_j - \boldsymbol{e}_{[s,f]}) - V^{q+1}_{\boldsymbol{\beta},j}(\boldsymbol{x}_j) \right] + \sum_{i\in\mathcal{N}\setminus\{j\}} \phi^q_i(S)\ \beta^q_{i,[s,f]\to j} \right\} \\
&\qquad\quad + \sum_{[s,f]\in\mathcal{F}} \mathbb{1}_{\{s\in[a,b],\ [s,f]\not\subseteq[a,b]\}}\, \lambda^q_{[s,f]} \times \\
&\qquad \max_{S\subseteq\mathcal{N}} \left\{ \phi^q_j(S) \left(\prod_{h=s}^{f} x_{j,h} \right) \left[\beta^q_{j,[s,f]\to j} + V^{q+1}_{\boldsymbol{\beta},j}(\boldsymbol{x}_j - \boldsymbol{e}_{[s,f]}) - V^{q+1}_{\boldsymbol{\beta},j}(\boldsymbol{x}_j) \right] + \sum_{i\in\mathcal{N}\setminus\{j\}} \phi^q_i(S)\ \beta^q_{i,[s,f]\to j} \right\} \\
&\overset{(a)}{=} \ \Theta^{q+1}_{\boldsymbol{\beta},j}(a,b) + \sum_{[s,f]\in\mathcal{F}} \mathbb{1}_{\{s\in[a,b],\ [s,f]\subseteq[a,b]\}}\, \lambda^q_{[s,f]} \times \\
&\qquad \max_{S\subseteq\mathcal{N}} \left\{ \phi^q_j(S) \left[\beta^q_{j,[s,f]\to j} + V^{q+1}_{\boldsymbol{\beta},j}(\boldsymbol{x}_j - \boldsymbol{e}_{[s,f]}) - V^{q+1}_{\boldsymbol{\beta},j}(\boldsymbol{x}_j) \right] + \sum_{i\in\mathcal{N}\setminus\{j\}} \phi^q_i(S)\ \beta^q_{i,[s,f]\to j} \right\} \\
&\qquad\quad + \sum_{[s,f]\in\mathcal{F}} \mathbb{1}_{\{s\in[a,b],\ [s,f]\not\subseteq[a,b]\}}\, \lambda^q_{[s,f]} \max_{S\subseteq\mathcal{N}\setminus\{j\}} \left\{ \sum_{i\in\mathcal{N}\setminus\{j\}} \phi^q_i(S)\ \beta^q_{i,[s,f]\to j} \right\},
\end{aligned} \tag{27}
$$

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 $\mathcal{I}(\boldsymbol{x}_j - \boldsymbol{e}_{[s,f]}) = \big(\mathcal{I}(\boldsymbol{x}_j) \setminus [a,b]\big) \cup \big\{[a,s-1],\ [f+1,b]\big\}$ for $[s,f] \subseteq [a,b]$. By definition of $\mathcal{K}(\boldsymbol{x}_j)$, $\mathcal{K}(\boldsymbol{x}_j - \boldsymbol{e}_{[s,f]}) = \mathcal{K}(\boldsymbol{x}_j) \cup \{s,\ldots,f\}$ So, by the induction hypothesis, we get

$$
\begin{aligned}
&V^{q+1}_{\boldsymbol{\beta},j}(\boldsymbol{x}_j - \boldsymbol{e}_{[s,f]]}) - V^{q+1}_{\boldsymbol{\beta},j}(\boldsymbol{x}_j) \\
&\qquad = \sum_{[a,b]\in\mathcal{I}(\boldsymbol{x}_j - \boldsymbol{e}_{[s,f]})} \Theta^{q+1}_{\boldsymbol{\beta},j}(a,b) \ + \sum_{\ell\in\mathcal{K}(\boldsymbol{x}_j - \boldsymbol{e}_{[s,f]})} \Psi^{q+1}_{\boldsymbol{\beta},j}(\ell) \ - \sum_{[a,b]\in\mathcal{I}(\boldsymbol{x}_j)} \Theta^{q+1}_{\boldsymbol{\beta},j}(a,b) \ - \sum_{\ell\in\mathcal{K}(\boldsymbol{x}_j)} \Psi^{q+1}_{\boldsymbol{\beta},j}(\ell) \\
&\qquad = \ \Theta^{q+1}_{\boldsymbol{\beta},j}(a,s-1) \ + \ \Theta^{q+1}_{\boldsymbol{\beta},j}(f+1,b) \ - \ \Theta^{q+1}_{\boldsymbol{\beta},j}(a,b) \ + \ \sum_{\ell=s}^{f} \Psi^{q+1}_{\boldsymbol{\beta},j}(\ell) \\
&\qquad \overset{(b)}{=} \ \Theta^{q+1}_{\boldsymbol{\beta},j}(a,s-1) \ + \ \Theta^{q+1}_{\boldsymbol{\beta},j}(f+1,b) \ - \ \Theta^{q+1}_{\boldsymbol{\beta},j}(a,b) \ + \ \Gamma^{q+1}_{\boldsymbol{\beta},j}(s,f),
\end{aligned} \tag{28}
$$

where (b) follows from the discussion right before the proof of the theorem, which shows that the value functions $\{\Gamma^q_{\boldsymbol{\beta},j} : q \in \mathcal{Q}\}$ and $\{\Psi^q_{\boldsymbol{\beta},j} : q \in \mathcal{Q}\}$ computed, respectively, through (14) and (24) satisfy the identity $\Gamma^{q+1}_{\boldsymbol{\beta},j}(s,f) = \sum_{\ell=s}^{f} \Psi^{q+1}_{\boldsymbol{\beta},j}(\ell)$. In this case, plugging (28) into (27), we equivalently express the right side of (25) as

$$\begin{aligned}
&\Theta^{q+1}_{\boldsymbol{\beta},j}(a,b) + \sum_{[s,f]\in\mathcal{F}} \mathbb{1}_{\{s\in[a,b],\ [s,f]\subseteq[a,b]\}}\, \lambda^q_{[s,f]} \times \\
&\qquad \max_{S\subseteq\mathcal{N}} \Bigg\{ \phi^q_j(S) \Big[\beta^q_{j,[s,f]\to j} + \Theta^{q+1}_{\boldsymbol{\beta},j}(a,s-1) \ + \ \Theta^{q+1}_{\boldsymbol{\beta},j}(f+1,b) \ - \ \Theta^{q+1}_{\boldsymbol{\beta},j}(a,b) \ + \ \Gamma^{q+1}_{\boldsymbol{\beta},j}(s,f) \Big] \\
&\qquad\qquad + \sum_{i\in\mathcal{N}\setminus\{j\}} \phi^q_i(S)\ \beta^q_{i,[s,f]\to j} \Bigg\} \\
&\qquad + \sum_{[s,f]\in\mathcal{F}} \mathbb{1}_{\{s\in[a,b],\ [s,f]\not\subseteq[a,b]\}}\, \lambda^q_{[s,f]} \max_{S\subseteq\mathcal{N}\setminus\{j\}} \Bigg\{ \sum_{i\in\mathcal{N}\setminus\{j\}} \phi^q_i(S)\ \beta^q_{i,[s,f]\to j} \Bigg\} \\
&\overset{(c)}{=} \ \Theta^q_{\boldsymbol{\beta},j}(a,b),
\end{aligned}$$

where (c) follows from (15). By the equality above, the right side of (25) is equal to $\Theta^q_{\boldsymbol{\beta},j}(a,b)$, establishing the equality in (25).

Second, we show that (26) holds for each $\ell \in \mathcal{K}(\boldsymbol{x}_j)$. Thus, we focus on the unavailable days $\ell \in \mathcal{K}(\boldsymbol{x}_j)$. For $\ell \in \mathcal{K}(\boldsymbol{x}_j)$, 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

$$\begin{aligned}
&\Psi^{q+1}_{\boldsymbol{\beta},j}(\ell) + \sum_{[s,f]\in\mathcal{F}} \mathbb{1}_{\{s=\ell\}}\, \lambda^q_{[s,f]} \times \\
&\qquad \max_{S\subseteq\mathcal{N}} \Bigg\{ \phi^q_j(S) \left(\prod_{h=s}^{f} x_{j,h} \right) \Big[\beta^q_{j,[s,f]\to j} + V^{q+1}_{\boldsymbol{\beta},j}(\boldsymbol{x}_j - \boldsymbol{e}_{[s,f]}) - V^{q+1}_{\boldsymbol{\beta},j}(\boldsymbol{x}_j) \Big] + \sum_{i\in\mathcal{N}\setminus\{j\}} \phi^q_i(S)\ \beta^q_{i,[s,f]\to j} \Bigg\} \\
&= \ \Psi^{q+1}_{\boldsymbol{\beta},j}(\ell) + \sum_{[s,f]\in\mathcal{F}} \mathbb{1}_{\{s=\ell\}}\, \lambda^q_{[s,f]} \max_{S\subseteq\mathcal{N}} \Bigg\{ \sum_{i\in\mathcal{N}\setminus\{j\}} \phi^q_i(S)\ \beta^q_{i,[s,f]\to j} \Bigg\} \\
&\overset{(d)}{=} \ \Psi^{q+1}_{\boldsymbol{\beta},j}(\ell) + \sum_{[s,f]\in\mathcal{F}} \mathbb{1}_{\{s=\ell\}}\, \lambda^q_{[s,f]} \max_{S\subseteq\mathcal{N}\setminus\{j\}} \Bigg\{ \sum_{i\in\mathcal{N}\setminus\{j\}} \phi^q_i(S)\ \beta^q_{i,[s,f]\to j} \Bigg\} \ \overset{(e)}{=} \ \Psi^q_{\boldsymbol{\beta},j}(\ell),
\end{aligned}$$

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. ■

Appendix H: Connection of the Upper Bound to Lagrangian Relaxation

In this section, we show that we can obtain the dynamic program in (12) through Lagrangian relaxation in the dynamic program in (1). In this way, we also relate our revenue allocations to Lagrange multipliers. In the maximization problem in (1), the decision variable S corresponds to the assortment of resources that we offer to a customer making a booking request for the interval

$[s,f]$. Using the decision variables $\{S_j : j \in \mathcal{N} \cup \{0\}\}$ instead of the decision variable S in the maximization problem in (1), we write the dynamic program in (1) equivalently as

$$J^q(\boldsymbol{x}) = \sum_{[s,f]\in\mathcal{F}} \lambda^q_{[s,f]} \max_{\substack{S_j \subseteq \mathcal{N} \\ \forall\, j \in \mathcal{N}\cup\{0\}}} \left\{ \sum_{j\in\mathcal{N}} \left(\prod_{\ell=s}^{f} x_{j,\ell}\right) \left\{ \phi^q_j(S_0)\, r_{j,[s,f]} + \phi^q_j(S_j) \Big[J^{q+1}(\boldsymbol{x} - \boldsymbol{e}_{j,[s,f]}) - J^{q+1}(\boldsymbol{x}) \Big] \right\} \right.$$
$$\left. : \left(\prod_{\ell=s}^{f} x_{j,\ell}\right) \phi^q_j(S_0) = \left(\prod_{\ell=s}^{f} x_{j,\ell}\right) \phi^q_j(S_i) \quad \forall\, i,j \in \mathcal{N} \right\} + J^{q+1}(\boldsymbol{x}). \quad (29)$$

We claim that the maximization problems on the right sides of (1) and (29) have the same optimal objective value. In particular, if S^* is an optimal solution to the maximization problem on the right side of (1), then setting $\hat{S}_j = S^*$ for all $j \in \mathcal{N} \cup \{0\}$ provides a feasible solution to the maximization problem on the right side of (29). Furthermore, the objective values of the two maximization problems at these two solutions match. On the other hand, if $\{S^*_j : j \in \mathcal{N} \cup \{0\}\}$ is an optimal solution to the maximization problem on the right side of (29), then setting $\hat{S} = S^*_0$ provides a feasible solution to the maximization problem on the right side of (1). Furthermore, noting that $\prod_{\ell=s}^{f} x_{j,\ell}\, \phi^q_j(\hat{S}) = \prod_{\ell=s}^{f} x_{j,\ell}\, \phi^q_j(S^*_0) = \prod_{\ell=s}^{f} x_{j,\ell}\, \phi^q_j(S^*_j)$ for all $j \in \mathcal{N}$ by the constraints in (29), the objective values of the two maximization problems at these two solutions match. Thus, the claim holds, which implies that the dynamic programs in (1) and (29) are equivalent to each other. Intuitively speaking, the dynamic program in (29) uses $N+1$ copies of the decision variable S in (1). These copies are given by $\{S_j : j \in \mathcal{N} \cup \{0\}\}$. Even though the dynamic program in (29) uses $N+1$ copies of the decision variable S in (1), the constraints in (29) ensure that the choice probabilities of all resources that are feasible to offer match under the different copies. We solve the maximization problem on the right side of (29) for each $[s,f] \in \mathcal{F}$. Thus, associating the Lagrange multipliers $\{\beta^q_{j,[s,f]\to i} : i,j \in \mathcal{N}\}$ with the constraints on the right side of (29), relaxing these constraints and arranging the terms, we obtain the dynamic program

$$\widetilde{J}^q_{\boldsymbol{\beta}}(\boldsymbol{x}) = \sum_{[s,f]\in\mathcal{F}} \lambda^q_{[s,f]} \max_{S_0 \in \mathcal{N}} \left\{ \sum_{j\in\mathcal{N}} \phi^q_j(S_0) \left(\prod_{\ell=s}^{f} x_{j,\ell}\right) \Big[r_{j,[s,f]} - \sum_{i\in\mathcal{N}} \beta^q_{j,[s,f]\to i} \Big] \right\}$$
$$+ \sum_{[s,f]\in\mathcal{F}} \lambda^q_{[s,f]} \sum_{j\in\mathcal{N}} \max_{S_j \in \mathcal{N}} \left\{ \phi^q_j(S_j) \left(\prod_{\ell=s}^{f} x_{j,\ell}\right) \Big[\beta^q_{j,[s,f]\to j} + \widetilde{J}^{q+1}_{\boldsymbol{\beta}}(\boldsymbol{x} - \boldsymbol{e}_{j,[s,f]}) - \widetilde{J}^{q+1}_{\boldsymbol{\beta}}(\boldsymbol{x}) \Big] \right.$$
$$\left. + \sum_{i\in\mathcal{N}\setminus\{j\}} \phi^q_i(S_j) \left(\prod_{\ell=s}^{f} x_{i,\ell}\right) \beta^q_{i,[s,f]\to j} \right\} + \widetilde{J}^{q+1}_{\boldsymbol{\beta}}(\boldsymbol{x}), \quad (30)$$

where we use the fact that once we relax the constraints in (29), the maximization problem on the right side of (29) decomposes by the elements of $\mathcal{N} \cup \{0\}$.

Note that we make the dependence of the value functions $\{\widetilde{J}^q_{\boldsymbol{\beta}} : q \in \mathcal{Q}\}$ in (30) on the Lagrange multipliers $\boldsymbol{\beta} = \{\beta^q_{i,[s,f]\to j} : i,j \in \mathcal{N},\ [s,f] \in \mathcal{F},\ q \in \mathcal{Q}\}$ explicit. The dynamic program in (30) is a relaxed version of the dynamic program in (29). It is a standard result that the value functions in (30) provide upper bounds on those in (29) for any choice of the Lagrange multipliers; see, for example, Adelman and Mersereau (2008). In other words, for any $\boldsymbol{\beta} \in \mathbb{R}^{N^2 \times |\mathcal{F}| \times |\mathcal{Q}|}$, we have $\widetilde{J}^q_{\boldsymbol{\beta}}(\boldsymbol{x}) \geq J^q(\boldsymbol{x})$ for each $\boldsymbol{x} \in \{0,1\}^{N \times T}$ and $q \in \mathcal{Q}$. Because of the terms $\left\{\prod_{\ell=s}^f x_{j,\ell} : j \in \mathcal{N}\right\}$ in (30), the value functions $\{\widetilde{J}^q_{\boldsymbol{\beta}} : q \in \mathcal{Q}\}$ can still be computationally difficult to compute. We carry out two additional forms of relaxation in (30), while making sure that we still obtain upper bounds on the value functions $\{J^q : q \in \mathcal{Q}\}$. First, recalling that we have $\widetilde{J}^q_{\boldsymbol{\beta}}(\boldsymbol{x}) \geq J^q(\boldsymbol{x})$ for any choice of the Lagrange multipliers, we choose our Lagrange multipliers such that $\sum_{i \in \mathcal{N}} \beta^q_{j,[s,f] \to i} = r_{j,[s,f]}$ for all $j \in \mathcal{N}$, $[s,f] \in \mathcal{F}$ and $q \in \mathcal{Q}$. In this case, the objective function of the first maximization problem in (30) becomes zero, so we can drop this maximization problem. Second, considering the second maximization problem in (30), by Lemma A.1, if $\beta^q_{i,[s,f] \to j} \leq 0$ for some $i \in \mathcal{N} \setminus \{j\}$ in the summation in the objective function of this maximization problem, then there exists an optimal solution to this maximization problem that does not include resource i. On the other hand, if we have $\beta^q_{i,[s,f] \to j} > 0$ for some $i \in \mathcal{N} \setminus \{j\}$, then $\prod_{\ell=s}^f x_{i,\ell}\, \beta^q_{i,[s,f] \to j} \leq \beta^q_{i,[s,f] \to j}$. Thus, if we drop the term $\prod_{\ell=s}^f x_{i,\ell}$ in the summation in the objective function of the second maximization problem in (30), then the optimal objective value of this maximization problem stays at least as large. In this case, focusing on Lagrange multipliers that satisfy $\sum_{i \in \mathcal{N}} \beta^q_{j,[s,f] \to i} = r_{j,[s,f]}$ for all $j \in \mathcal{N}$, $[s,f] \in \mathcal{F}$ and $q \in \mathcal{Q}$ and dropping the term $\prod_{\ell=s}^f x_{i,\ell}$ in the summation in the objective function of the second maximization problem in (30), we get the dynamic program

$$\Theta^q_{\boldsymbol{\beta}}(\boldsymbol{x}) = \sum_{[s,f] \in \mathcal{F}} \lambda^q_{[s,f]} \sum_{j \in \mathcal{N}} \max_{S_j \in \mathcal{N}} \left\{ \phi^q_j(S_j) \left(\prod_{\ell=s}^f x_{j,\ell} \right) \left[\beta^q_{j,[s,f] \to j} + \Theta^{q+1}_{\boldsymbol{\beta}}(\boldsymbol{x} - \boldsymbol{e}_{j,[s,f]}) - \Theta^{q+1}_{\boldsymbol{\beta}}(\boldsymbol{x}) \right] \right.$$
$$\left. + \sum_{i \in \mathcal{N} \setminus \{j\}} \phi^q_i(S_j)\, \beta^q_{i,[s,f] \to j} \right\} + \Theta^{q+1}_{\boldsymbol{\beta}}(\boldsymbol{x}). \quad (31)$$

By the discussion in this paragraph, the value functions $\{\Theta^q_{\boldsymbol{\beta}} : q \in \mathcal{Q}\}$ in (31) provide upper bounds on the value functions $\{\widetilde{J}^q_{\boldsymbol{\beta}} : q \in \mathcal{Q}\}$ in (30), as long as $\sum_{i \in \mathcal{N}} \beta^q_{j,[s,f] \to i} = r_{j,[s,f]}$.

Summing up the development so far, if the Lagrange multipliers satisfy $\sum_{i \in \mathcal{N}} \beta^q_{j,[s,f] \to i} = r_{j,[s,f]}$ for all $j \in \mathcal{N}$, $[s,f] \in \mathcal{F}$ and $q \in \mathcal{Q}$, then we have $\Theta^q_{\boldsymbol{\beta}}(\boldsymbol{x}) \geq \widetilde{J}^q_{\boldsymbol{\beta}}(\boldsymbol{x}) \geq J^q(\boldsymbol{x})$ for each $\boldsymbol{x} \in \{0,1\}^{N \times T}$ and $q \in \mathcal{Q}$. Therefore, we can obtain an upper bound on the value functions $\{J^q : q \in \mathcal{Q}\}$ in (1) by using the value functions $\{\Theta^q_{\boldsymbol{\beta}} : q \in \mathcal{Q}\}$ in (31). Furthermore, we obtain the dynamic program in (31) by using Lagrangian relaxation on the dynamic program in (1). In the next lemma, we show that the dynamic program in (31) is equivalent to the one in (12). Thus, we can obtain the dynamic program in (12), which we use to compute an upper bound on the optimal total expected revenue in Section 6, by using Lagrangian relaxation on the dynamic program in (1). In the next lemma, recall that we use $\boldsymbol{x}_j = (x_{j,1}, \ldots, x_{j,T}) \in \{0,1\}^T$ to capture the state of resource j, where $x_{j,\ell} = 1$ if and only if resource j is available for use on day ℓ.

Lemma H.1 (Equivalence of Dynamic Programs) *Letting the value functions* $\{V^q_{\boldsymbol{\beta},j} : q \in \mathcal{Q}\}$ *be computed through* (12) *and the value functions* $\{\Theta^q_{\boldsymbol{\beta}} : q \in \mathcal{Q}\}$ *be computed through* (31), *for each* $\boldsymbol{x} = (\boldsymbol{x}_j : j \in \mathcal{N}) \in \{0,1\}^{N \times T}$ *and* $q \in \mathcal{Q}$, *we have*

$$\Theta^q_{\boldsymbol{\beta}}(\boldsymbol{x}) = \sum_{j \in \mathcal{N}} V^q_{\boldsymbol{\beta},j}(\boldsymbol{x}_j).$$

Proof: We show the result by using induction over the time periods. At time period $Q+1$, we have $\Theta^{Q+1}_{\boldsymbol{\beta}} = 0 = V^{Q+1}_{\boldsymbol{\beta},j}$, 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 argument, we have $\Theta^{q+1}_{\boldsymbol{\beta}}(\boldsymbol{x}) = \sum_{j \in \mathcal{N}} V^{q+1}_{\boldsymbol{\beta},j}(\boldsymbol{x}_j)$, so $\Theta^{q+1}_{\boldsymbol{\beta}}(\boldsymbol{x} - \boldsymbol{e}_{j,[s,f]}) - \Theta^{q+1}_{\boldsymbol{\beta}}(\boldsymbol{x}) = V^{q+1}_{\boldsymbol{\beta},j}(\boldsymbol{x}_j - \boldsymbol{e}_{[s,f]}) - V^{q+1}_{\boldsymbol{\beta},j}(\boldsymbol{x}_j)$, in which case, using the dynamic program in (31), we obtain

$$\begin{aligned}\Theta^q_{\boldsymbol{\beta}}(\boldsymbol{x}) \;=\; \sum_{[s,f] \in \mathcal{F}} \lambda^q_{[s,f]} \sum_{j \in \mathcal{N}} \max_{S_j \in \mathcal{N}} \Bigg\{ \phi^q_j(S_j) \left(\prod_{\ell=s}^{f} x_{j,\ell} \right) \Big[\beta^q_{j,[s,f] \to j} + V^{q+1}_{\boldsymbol{\beta},j}(\boldsymbol{x}_j - \boldsymbol{e}_{[s,f]}) - V^{q+1}_{\boldsymbol{\beta},j}(\boldsymbol{x}_j) \Big] \\ + \sum_{i \in \mathcal{N} \setminus \{j\}} \phi^q_i(S_j)\, \beta^q_{i,[s,f] \to j} \Bigg\} + \sum_{j \in \mathcal{N}} V^{q+1}_{\boldsymbol{\beta},j}(\boldsymbol{x}_j) \;=\; \sum_{j \in \mathcal{N}} V^q_{\boldsymbol{\beta},j}(\boldsymbol{x}_j),\end{aligned}$$

where the last equality follows by arranging the terms and using (12). Thus, the result holds at time period q as well, completing the induction argument. ■

By the lemma above, the dynamic program in (31) decomposes by the resources and the dynamic program that we solve for each resource j corresponds to the one in (12). Noting that we obtained the dynamic program in (31) by using Lagrangian relaxation on the dynamic program in (1), the dynamic program that we solve in Section 6 to compute an upper bound on the optimal total expected revenue can be obtained by using Lagrangian relaxation on the dynamic program in (1). The revenue allocations $\{\beta^q_{i,[s,f]\to j} : i,j \in \mathcal{N},\ [s,f] \in \mathcal{F},\ q \in \mathcal{Q}\}$ in Section 6 correspond to the Lagrange multipliers employed when using Lagrangian relaxation on (1).

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

In Baek and Ma (2021), 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 refer to the policy proposed by Baek and Ma (2021) as the constraint splitting policy. The performance guarantee for the constraint splitting policy applies to general revenue management

Param. $(\mathsf{D}_{\max}, \rho, \delta)$	Total Exp. Revenue LINR	POLR	LPR	COS1	COS5
$(6, 1.0, 0.9)$	78.00	78.17	78.55	67.57	64.82
$(6, 1.0, 0.7)$	83.41	83.54	83.87	71.83	68.97
$(6, 1.2, 0.9)$	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
$(6, 1.4, 0.9)$	75.51	76.04	75.77	67.66	65.62
$(6, 1.4, 0.7)$	80.83	80.22	81.01	72.13	67.84
$(8, 1.0, 0.9)$	79.13	78.41	79.40	63.34	59.94
$(8, 1.0, 0.7)$	78.15	77.01	78.64	59.47	57.66
$(8, 1.2, 0.9)$	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
$(8, 1.4, 0.9)$	81.04	79.34	80.41	66.64	65.12
$(8, 1.4, 0.7)$	78.26	76.56	78.33	63.52	63.79
$(10, 1.0, 0.9)$	70.53	70.43	71.54	54.67	54.70
$(10, 1.0, 0.7)$	77.11	74.66	76.79	58.87	54.54
$(10, 1.2, 0.9)$	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
$(10, 1.4, 0.9)$	80.92	76.82	80.46	61.49	55.58
$(10, 1.4, 0.7)$	80.51	77.52	80.34	62.32	61.72
Avg.	77.43	76.41	77.90	63.00	60.86

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

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 $\frac{1}{1+\mathsf{D}_{\max}}$. Our polynomial value function approximations improve upon this performance guarantee when $\mathsf{D}_{\min} > 1$. 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, 27.24%, 25.55%, and 28.02%. Thus, exploiting the structure of our revenue management problem, our rollout policies provide substantial improvements over COS5. Similarly, the performance of LINR, POLR and LPR is better than that of COS1 by, respectively, 22.91%, 21.29% and 23.66%, on average. 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, 24.23%, 14.66%, and 20.46%, on average.

Param.	Upp.		Total Exp. Revenue										
$(\mathsf{D}_{\max},\rho,\delta)$	Bnd.	$\mathsf{Z}^*_{\mathsf{LP}}$	LIN1	POL1	LP1	LIN5	POL5	LP5	LINR	POLR	LPR	GUB	DEC
$(6,1.0,0.9)$	3,337	3,990	2,485	2,376	2,449	2,570	2,421	2,515	2,603	2,608	2,621	2,616	2,470
$(6,1.0,0.7)$	1,195	1,526	964	900	934	983	919	961	996	998	1,002	1,015	962
$(6,1.2,0.9)$	2,525	2,914	2,001	1,899	1,944	2,043	1,903	1,970	2,047	2,027	2,048	2,016	1,951
$(6,1.2,0.7)$	4,120	4,615	2,891	2,825	2,824	2,936	2,877	2,871	3,008	3,010	3,048	3,001	2,788
$(6,1.4,0.9)$	6,029	7,138	4,316	4,297	4,249	4,460	4,388	4,286	4,553	4,585	4,568	4,639	4,344
$(6,1.4,0.7)$	3,674	4,580	2,826	2,597	2,760	2,897	2,683	2,864	2,970	2,947	2,976	3,020	2,832
$(8,1.0,0.9)$	3,735	4,749	2,836	2,510	2,720	2,897	2,565	2,807	2,955	2,928	2,965	3,029	2,886
$(8,1.0,0.7)$	3,365	4,227	2,483	2,324	2,447	2,546	2,348	2,446	2,630	2,591	2,646	2,615	2,556
$(8,1.2,0.9)$	4,607	5,788	3,468	3,049	3,404	3,600	3,192	3,497	3,604	3,586	3,653	3,678	3,492
$(8,1.2,0.7)$	3,663	4,232	2,579	2,407	2,456	2,644	2,436	2,511	2,604	2,604	2,690	2,648	2,506
$(8,1.4,0.9)$	4,150	5,179	3,140	2,829	3,107	3,228	2,909	3,145	3,363	3,293	3,337	3,374	3,191
$(8,1.4,0.7)$	5,226	6,507	3,811	3,588	3,826	3,995	3,669	3,861	4,090	4,001	4,093	4,084	3,889
$(10,1.0,0.9)$	6,884	7,523	4,615	4,526	4,300	4,737	4,626	4,410	4,856	4,849	4,925	4,594	4,494
$(10,1.0,0.7)$	5,317	6,387	3,898	3,552	3,780	3,977	3,656	3,845	4,100	3,969	4,082	4,017	3,895
$(10,1.2,0.9)$	5,259	5,986	3,608	3,291	3,484	3,719	3,336	3,574	3,886	3,757	3,909	3,752	3,684
$(10,1.2,0.7)$	5,403	6,102	3,780	3,596	3,621	3,777	3,700	3,727	3,949	3,939	4,054	3,864	3,715
$(10,1.4,0.9)$	5,708	7,316	4,357	3,562	4,221	4,496	3,680	4,346	4,619	4,385	4,593	4,702	4,393
$(10,1.4,0.7)$	3,414	4,421	2,546	2,315	2,555	2,586	2,345	2,578	2,749	2,647	2,743	2,783	2,531

Table EC.2 Absolute values of the upper bounds and total expected revenues for all benchmark policies.

Appendix J: Absolute Values of Upper Bounds and Total Expected Revenues

In Table 1, we give the gap between the upper bound from our approach and the upper bound from the linear program in (11). Similarly, in Table 2, we give the total expected revenues of the benchmark policies as a percentage of the upper bound. In particular, these tables do not give the absolute values of the upper bounds or total expected revenues. To ensure that our computational results are transparent, in Table EC.2, we give the absolute values of all upper bounds and total expected revenues for all of our test problems. In this table, the first and second columns, respectively, give the upper bounds obtained by using the dynamic program in (12) and the linear programming approximation in (11). The last eleven columns give the total expected revenues obtained by the benchmark policies in Section 7.1. Thus, the entries in Table 1 correspond to the percent gap between the entries in the first and second columns in Table EC.2, whereas the entries in each column in Table 2 correspond to the entries in each of last eleven columns in Table EC.2 expressed as a percentage of the entries in the first column in Table EC.2.

Appendix K: Heuristic Modification of the Resource Based Static Policy

In our discussion of the resource based static policy, we intuitively interpreted the fraction $\frac{\mathbb{1}_{\{\ell\in[s,f]\}}}{f-s+1}$ in (7) as a way of spreading the net revenue from a booking request for interval $[s,f]$ evenly over each day $\ell\in\{s,\dots,f\}$. This interpretation is merely a heuristic and the main reason that this fraction appears in (7) is algebraic, resting on the the fact that we have the relationship $\prod_{\ell=s}^{f} x_{i,\ell}\geq\frac{\left(\sum_{\ell=s}^{f}x_{i,\ell}\right)-(f-s)}{1+f-s}$ for any $\boldsymbol{x}\in\{0,1\}^{N\times T}$, as established in Lemma B.1. In particular, we use this relationship in the proof of Proposition B.5 to lower bound the total expected revenue of the resource based static policy. Even though the main motivation for the fraction $\frac{\mathbb{1}_{\{\ell \in [s,f]\}}}{f-s+1}$ in (7) is algebraic, an interesting question is whether we can use different fractions in (7) to come up with variants of the resource based static policy. These variants would be heuristics and would not necessarily inherit the performance guarantee of $\frac{1}{1+2\,\mathsf{D}_{\max}}$. We experimented with a variant of the resource based static policy, where we replace the fraction $\frac{\mathbb{1}_{\{\ell \in [s,f]\}}}{f-s+1}$ with $\frac{\mathbb{1}_{\{\ell \in [s,f]\}} \zeta_{i,\ell}}{\sum_{t=s}^{f} \zeta_{i,t}}$, where $\zeta_{i,\ell}$ is, roughly speaking, the weight that we attach to resource i on day ℓ. Letting $\{\hat{\mu}_{i,\ell} : i \in \mathcal{N},\ \ell \in \mathcal{T}\}$ be the optimal values of the dual variables associated with the first constraint in problem (11), we can use $\hat{\mu}_{i,\ell}$ to capture the opportunity cost of the capacity for resource i on day ℓ. Thus, we set the weights $\{\zeta_{i,\ell} : i \in \mathcal{N},\ \ell \in \mathcal{T}\}$ as $\zeta_{i,\ell} = \hat{\mu}_{i,\ell}$ for all $i \in \mathcal{N}$ and $\ell \in \mathcal{T}$. We refer to this policy as the resource based static policy with weights. Once again, we emphasize that the resource based static policy with weights is a heuristic extension and does not necessarily have a performance guarantee, because we cannot use the fraction $\frac{\mathbb{1}_{\{\ell \in [s,f]\}} \zeta_{i,\ell}}{\sum_{t=s}^{f} \zeta_{i,t}}$ to lower bound the total expected revenue of the resource based static policy with weights, as is done in the proof of Proposition B.5.

In our implementation of the resource based static policy with weights, we compute the opportunity costs $\{\eta^q_{i,\ell} : i \in \mathcal{N},\ \ell \in \mathcal{T},\ q \in \mathcal{Q}\}$ once and five times over the selling horizon, yielding the benchmark policies LW1 and LW5. We also perform rollout on the resource based static policy with weights, yielding the benchmark policy LWR. In Table EC.3, we compare the performance of the resource based static policy with weights against the performance of the original resource based static policy. Recall that we compute the opportunity costs $\{\eta^q_{i,\ell} : i \in \mathcal{N},\ \ell \in \mathcal{T},\ q \in \mathcal{Q}\}$ for the original resource based static policy once and five times over the selling horizon, yielding the benchmark policies LIN1 and LIN5. We also perform rollout on the original resource based static policy, yielding the benchmark policy LINR. We express all total expected revenues as a percentage of the upper bound on the optimal total expected revenue. Our results indicate that LW1 and LW5 lag behind LIN1 and LIN5 slightly, but once we perform rollout, the resource based static policy with weights is comparable to the original resource based static policy, even though the resource based static policy with weights does not necessarily have a performance guarantee.

Appendix L: 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 (2021).

Parameter Estimation: Recalling that the values of $\{\theta_k : k = 1, \ldots, 171\}$ 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 $\{\theta_k : k = 1, \ldots, 171\}$. The length of stay for a customer ranges between one and eight days, so we need to estimate eight parameters for $\{\eta_d : d = 1, \ldots, 8\}$.

Param.	Total Exp. Revenue					
$(D_{\max}, \rho, \delta)$	LIN1	LW1	LIN5	LW5	LINR	LWR
$(6, 1.0, 0.9)$	74.46	74.92	77.02	75.03	78.00	77.95
$(6, 1.0, 0.7)$	80.66	79.46	82.33	78.69	83.41	83.41
$(6, 1.2, 0.9)$	79.23	77.33	80.89	77.55	81.05	79.62
$(6, 1.2, 0.7)$	70.17	70.69	71.27	71.60	73.00	73.95
$(6, 1.4, 0.9)$	71.58	70.26	73.98	72.03	75.51	75.53
$(6, 1.4, 0.7)$	76.93	75.62	78.84	77.72	80.83	80.90
$(8, 1.0, 0.9)$	75.93	75.13	77.58	74.07	79.13	79.22
$(8, 1.0, 0.7)$	73.78	71.99	75.65	71.80	78.15	78.34
$(8, 1.2, 0.9)$	75.27	74.64	78.14	74.93	78.23	78.62
$(8, 1.2, 0.7)$	70.39	69.67	72.16	70.54	71.07	72.54
$(8, 1.4, 0.9)$	75.66	75.07	77.78	74.35	81.04	81.18
$(8, 1.4, 0.7)$	72.94	71.97	76.44	72.89	78.26	78.02
$(10, 1.0, 0.9)$	67.04	67.43	68.81	65.38	70.53	70.29
$(10, 1.0, 0.7)$	73.32	71.57	74.80	71.04	77.11	76.74
$(10, 1.2, 0.9)$	68.61	69.59	70.72	68.07	73.89	72.44
$(10, 1.2, 0.7)$	69.95	69.66	69.90	68.75	73.09	74.03
$(10, 1.4, 0.9)$	76.34	75.47	78.76	74.57	80.92	80.90
$(10, 1.4, 0.7)$	74.57	73.78	75.74	73.98	80.51	78.19
Avg.	73.71	73.01	75.60	72.94	77.43	77.33

Table EC.3 Total expected revenues obtained by the resource based static policy with weights.

Lastly, there are six rooms, which implies that we need to estimate the six preference weights $\{v_i : i = 1, \ldots, 6\}$. 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

Load Fact.	Total Exp. Revenue				
	LINR	POLR	LPR	COS1	COS5
0.8	83.96	84.43	83.00	78.07	78.22
1.0	86.91	87.26	85.88	81.97	81.93
1.2	88.18	88.70	87.16	82.93	83.64
1.4	89.38	89.36	88.31	83.86	84.45
1.6	90.18	89.69	88.85	85.18	84.77
Avg.	87.72	87.89	86.64	82.40	82.60

Table EC.4 Total expected revenues obtained by the constraint splitting policies.

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 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 I, we compute the parameters of the constraint splitting policy in Baek and Ma (2021) once and five times over the selling horizon, yielding the policies COS1 and COS5. In Table EC.4, 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 provides noticeable improvements over COS1 and COS5.

Appendix M: Test Problems with a Single Resource

We test the performance of our policies on test problems with a single resource. Our experimental setup follows the one in Section 7.1 except for the fact that we have a single resource, so that $N=1$. This resource is available for use during the days indexed by $\mathcal{T}=\{1,\ldots,70\}$, corresponding to 10 weeks in the booking horizon. Customers arrive over time periods $\mathcal{Q}=\{1,\ldots,700\}$. Recalling that $\mathsf{D}_{\max}$ is the maximum number of days that the resource can be booked for, ρ is the load factor, and δ is the discount parameter for booking the resource on a weekday, we vary the parameters $\mathsf{D}_{\max}\in\{3,6\}$, $\rho\in\{1.2,1.6,2.0\}$ and $\delta\in\{0.7,0.9\}$ to obtain 12 parameter configurations for our test problems. Since we have one resource, the approach described in Section 6 computes the optimal total expected revenue for our test problems. In particular, recalling that $\boldsymbol{e}'\in\{0,1\}^T$ is the vector of all ones and indexing the single resource by j, if we set the revenue allocations $\hat{\boldsymbol{\beta}}=\{\hat{\beta}^q_{j,[s,f]\to j}:[s,f]\in\mathcal{F},\ q\in\mathcal{Q}\}$ as $\hat{\beta}^q_{j,[s,f]\to j}=r_{j,[s,f]}$ and compute the value functions $\{V^q_{\hat{\boldsymbol{\beta}},j}:q\in\mathcal{Q}\}$ through the dynamic program in (12), then $V^1_{\hat{\boldsymbol{\beta}},j}(\boldsymbol{e}')$ is the optimal total expected revenue. We give our computational results in Table EC.5. The first column in this table gives the parameter

Param.	Gap in	Total Exp. Revenue								
$(\mathsf{D}_{\max}, \rho, \delta)$	Bounds	LIN1	POL1	LP1	LIN5	POL5	LP5	LINR	POLR	LPR
$(3, 1.2, 0.9)$	47.79	94.42	99.87	95.09	94.45	99.89	94.40	100.00	100.00	100.00
$(3, 1.2, 0.7)$	47.77	94.52	100.00	96.73	94.44	100.00	95.52	100.00	100.00	100.00
$(3, 1.6, 0.9)$	44.37	97.60	99.38	94.88	97.51	99.62	94.60	99.77	100.00	99.91
$(3, 1.6, 0.7)$	44.17	96.64	98.67	95.08	96.54	98.85	94.82	99.61	99.75	99.43
$(3, 2.0, 0.9)$	41.81	97.09	97.75	85.84	96.99	98.04	92.13	99.11	99.84	99.64
$(3, 2.0, 0.7)$	41.71	97.14	98.02	86.78	96.72	98.25	94.31	99.88	100.00	100.00
$(6, 1.2, 0.9)$	50.39	95.66	95.87	91.40	94.89	96.44	88.05	100.00	100.00	100.00
$(6, 1.2, 0.7)$	50.27	94.66	94.99	91.41	94.11	95.34	87.93	99.84	99.63	99.76
$(6, 1.6, 0.9)$	47.84	97.21	92.78	94.62	96.74	93.57	93.94	100.00	100.00	100.00
$(6, 1.6, 0.7)$	47.63	95.96	92.08	93.85	95.34	92.97	92.63	99.67	99.59	99.86
$(6, 2.0, 0.9)$	45.14	96.55	89.82	91.69	96.01	90.84	92.80	100.00	100.00	100.00
$(6, 2.0, 0.7)$	45.02	95.54	87.95	90.34	94.88	89.36	91.74	99.28	99.02	99.35
Avg.	46.16	96.08	95.64	92.31	95.72	96.14	92.74	99.76	99.82	99.83

Table EC.5 Computational results for the test problems with a single resource.

configuration for each test problem. The second column shows the percent gap between the upper bounds obtained by using the dynamic program in (12) and the linear program in (11). Since we can obtain the optimal total expected revenue through the dynamic program in (12), this column shows how far the upper bound provided by the linear program in (11) is from the optimal total expected revenue. The remaining columns give the total expected revenues obtained by the benchmark policies expressed as a percentage of the optimal total expected revenue.

Our results indicate that the total expected revenues obtained by our rollout policies are within only a fraction of a percent of the optimal total expected revenue even in the worst case. For many test problems, our rollout policies are optimal. One may conjecture that the rollout policies are provably optimal when there is a single resource, but it is not difficult to construct counterexamples to show that this conjecture is not correct. The performance of the policies that do not use rollout may lag behind the optimal total expected revenue by as much as 14.16%. Lastly, the upper bounds provided by the linear program in (11) can be rather loose. The gaps between these upper bounds and the optimal total expected revenue can reach 50.39%.

Appendix N: Heuristic Extension to Multiple Units of Resource Capacity

The performance guarantee for the resource based static policy, as well as our results for rolling out a static policy, use the fact that each resource has unit capacity. It appears to be nontrivial to extend these results to the case with multiple units of capacity for a resource. In this section, we give a heuristic modification of the resource based static policy when there are multiple units of a resource and test this modification in computational experiments. In particular, we consider the case where there are $C_{i,\ell}$ units of resource i available on day ℓ. In this case, we compute the opportunity costs $\{\eta^q_{i,\ell} : i \in \mathcal{N},\ \ell \in \mathcal{T},\ q \in \mathcal{Q}\}$ for the resource based static policy after replacing $\phi^q_i(A^q_{[s,f]})$ in (7) with $\frac{1}{C_{i,\ell}}\, \phi^q_i(A^q_{[s,f]})$. The heuristic reasoning behind this modification is that $\eta^q_{i,\ell}$

characterizes the opportunity cost of a unit of capacity for resource i on day ℓ. Given that we offer the assortment $A^q_{[s,f]}$ at time period q, a customer chooses resource i with probability $\phi^q_i(A^q_{[s,f]})$. Because there are $C_{i,\ell}$ units of resource i available on day ℓ, each unit of resource faces demand with probability $\frac{1}{C_{i,\ell}}\,\phi^q_i(A^q_{[s,f]})$. At time period q, there actually may not be $C_{i,\ell}$ units of capacity of resource i available on day ℓ because of the booking requests at the previous time periods, so this reasoning is a heuristic. Also, Lemma B.1 is critical in establishing a performance guarantee for the resource based static policy and it holds only when the capacity of each resource is one.

Our computational experiments are for a boutique hotel. The set of resources corresponds to a set of room types. We have $N=3$ room types. We have K units of each room type available on each day, so we have $C_{i,\ell}=K$ for all $i\in\mathcal{N}$ and $\ell\in\mathcal{T}$. We generate our test problems by using the same approach in Section 7.1. In all of our computational experiments, we set the discount parameter as $\delta=0.7$ and the maximum duration of stay parameter as $\mathsf{D}_{\max}=6$. Noting that the total available capacity that we have in all rooms on all days is KNT, the load factor parameter satisfies $\rho=\frac{\lambda^q_{[s,f]}}{\gamma^q_{[s,f]}}\,\frac{\mathsf{Demand}}{KNT}$ for each interval $[s,f]$. We use the same approach in Section 7.1 to generate the arrival probabilities for the booking requests for different intervals and the parameters of the multinomial logit model. Varying the parameters $\rho\in\{1.2,1.6,2.0\}$ and $K=\{1,3,5\}$, we obtain nine parameter configurations for our test problems.

We compute the opportunity costs $\{\eta^q_{i,\ell}: i\in\mathcal{N},\ \ell\in\mathcal{T},\ q\in\mathcal{Q}\}$ for the resource based static policy once and five times over the selling horizon, yielding the benchmark policies LIN1 and LIN5. In the linear program in (11), we replace the right side of the first constraint with $\{C_{i,\ell}: i\in\mathcal{N},\ \ell\in\mathcal{T}\}$, in which case, letting $\{\hat{\mu}_{i,\ell}: i\in\mathcal{N},\ \ell\in\mathcal{T}\}$ be the optimal values of the dual variables associated with this constraint, we construct a benchmark policy by using $\hat{\mu}_{i,\ell}$ as the opportunity cost of the unit of capacity of resource i on day ℓ. We compute the opportunity costs once and five times over the selling horizon, yielding the benchmark policies LP1 and LP5. We also use the constraint splitting policy in Baek and Ma (2021), where we compute the parameters of this policy once and five times over the selling horizon, yielding the benchmark policies COS1 and COS5. Lastly, we implement the dynamic programming decomposition method, yielding the benchmark policy DEC.

We give our computational results in Table EC.6. The first column in this table gives the parameter configuration for each test problem. The remaining columns give the total expected revenues obtained by the benchmark policies expressed as a percentage of the upper bound on the optimal total expected revenue. We use the optimal objective value of the linear program in (11) as an upper bound on the optimal total expected revenue, because we can efficiently implement the approach in Section 6 to obtain an upper bound only when we have unit capacities for all of the resources. Our results indicate that LIN5, LP5 and DEC are the strongest benchmark policies. For

Param.	Total Exp. Revenue						
(ρ, K)	LIN1	LP1	COS1	LIN5	LP5	COS5	DEC
$(1.2, 1)$	71.97	73.69	70.34	75.34	72.87	67.98	72.83
$(1.6, 1)$	79.31	72.53	77.67	80.81	78.79	73.29	81.14
$(2.0, 1)$	80.47	76.54	76.33	82.97	80.96	74.34	80.85
$(1.2, 5)$	75.61	77.13	66.62	79.53	78.95	64.29	78.56
$(1.6, 5)$	75.07	73.87	74.26	77.66	79.03	70.20	78.49
$(2.0, 5)$	76.40	75.94	75.01	80.57	80.90	70.53	79.68
$(1.2, 10)$	70.58	72.63	62.14	73.12	75.14	58.04	74.75
$(1.6, 10)$	52.34	55.49	49.44	55.47	57.35	46.25	56.76
$(2.0, 10)$	43.19	43.19	43.48	45.58	46.38	40.14	45.86
Avg.	69.44	69.00	66.14	72.34	72.26	62.78	72.10

Table EC.6 Total expected revenues obtained by the benchmark policies under multiple units of resource capacity.

this problem class, the constraint splitting policy in Baek and Ma (2021) continues to provide a performance guarantee of $\frac{1}{1+\mathsf{D}_{\max}}$, but our modification of the resource based static policy is a heuristic. Nevertheless, LIN1 and LIN5 perform noticeably better than COS1 and COS5.