# Product-Based Approximate Linear Programs for Network Revenue Management

Rui Zhang
Leeds School of Business, University of Colorado, 995 Regent Drive, Boulder, CO 80309
rui.zhang@colorado.edu

Saied Samiedaluie
Alberta School of Business, University of Alberta, Edmonton, Alberta T6G 2R6, Canada
samiedal@ualberta.ca

Dan Zhang
Leeds School of Business, University of Colorado, 995 Regent Drive, Boulder, CO 80309
dan.zhang@colorado.edu

The approximate linear programming approach has received significant attention in the network revenue management literature. A popular approximation in the existing literature is separable piecewise linear (SPL) approximation, which estimates the value of each unit of each resource over time. SPL approximation can be used to construct *resource-based* bid-price policies. In this paper, we propose a *product-based* SPL approximation. The coefficients of the product-based SPL approximation can be interpreted as each product's revenue contribution to the value of each unit of each resource in a given period. We show that the resulting approximate linear program (ALP) admits compact reformulations, like its resource-based counterpart. Furthermore, the new approximation allows us to derive a set of valid inequalities to (i) speed up the computation and (ii) select optimal solutions to construct more effective policies. We conduct an extensive numerical study to illustrate our results. In a set of 192 problem instances, bid-price policies based on the new approximation generate higher expected revenues than resource-based bid-price policies, with an average revenue lift of 0.72% and a maximum revenue lift of 5.3%. In addition, the new approximation can be solved 1.42 times faster than the resource-based approximation and shows better numerical stability. The valid inequalities derived from the new approximation further improve the computational performance and are critical for achieving additional gains in the expected revenue. The policy performance is competitive compared with the dynamic programming decomposition method, which is the strongest heuristic known in the literature.

*Key words*: Approximate Dynamic Programming, Optimal Control, Transportation, Yield Management

## 1. Introduction

Network revenue management (NRM) entails managing a network of resources with limited capacities, which are consumed by different products over time. Product requests arrive over a finite selling horizon. The objective is to maximize the total expected revenue by accepting or rejecting these requests. This problem can be formulated as a dynamic program, where the state represents the remaining capacities of the resources (Talluri and van Ryzin 2004). It can be shown that an optimal policy accepts a booking request, as long as there are sufficient resources and the price exceeds the marginal cost. However, the dynamic programming formulation suffers from the well-known "curse of dimensionality." Therefore, practical solution methods resort to approximations and heuristics.

The approximate linear programming approach has received significant attention in the literature (de Farias and Van Roy 2003, de Farias and Van Roy 2004). The main idea of this approach is to represent the value function of the dynamic program by a weighted sum of a collection of basis functions, which dramatically reduces the number of variables. Building on this idea, Adelman (2007) introduces a solution framework for the NRM problem by approximating the value function as an affine function of the state. He shows that the resulting affine approximate linear program (ALP) produces a tighter upper bound on the total expected revenue than the widely used deterministic linear program (DLP) (Talluri and van Ryzin 1998, Cooper 2002). The coefficients of the affine approximation, interpreted as the marginal values of the resources in each period, can be used to construct a dynamic time-dependent bid-price policy, which is shown to be superior to the static bid-price policies obtained from DLP (Williamson 1992, Talluri and van Ryzin 1998, Cooper 2002).

A stream of follow-up work to Adelman (2007) considers stronger functional approximations that produce tighter upper bounds and possibly stronger heuristic policies. A popular approximation is the separable piecewise linear (SPL) approximation (Farias and Van Roy 2007, Meissner and Strauss 2012). Topaloglu (2009) proposes a Lagrangian relaxation approach to the NRM problem, which is subsequently shown to be equivalent to the SPL approximation (Kunnumkal and Talluri 2016, Vossen and Zhang 2015b). In this paper, we refer to the SPL approximation as the *resource-based* SPL approximation to differentiate it from an alternative approximation that we propose.

The coefficients of the resource-based SPL approximation can be interpreted as the marginal values of resources in each period and immediately imply a dynamic policy that is both time and resource-level dependent. The policy builds on the widely used idea of bid-price control, where a product request is accepted as long as the price exceeds the sum of the values of the utilized resources (Talluri and van Ryzin 1998). We call such policies *resource-based* bid-price policies. Talluri and van Ryzin (1998) use an example to show that the resource-based bid-price policies ignore the network effect among products sharing the same resources, and are therefore suboptimal in general.

We propose an alternative *product-based* SPL approximation to alleviate the shortcomings of the resource-based SPL approximation and its corresponding policies. We also construct a new policy based on the proposed approximation, which we call the *product-based* bid-price policy. The novel feature of the new policy is that it keeps track of the resource levels and the feasibility of offering each product. Thus, it takes the network effect into account when deciding whether to accept a product request. The new policy is optimal for the example from Talluri and van Ryzin

(1998). A numerical study with 192 problem instances shows that the product-based bid-price policy can generate higher expected revenues than the resource-based bid-price policy, with an average revenue gain of 0.72% and a maximum of 5.3%. The magnitude of the revenue improvement is quite substantial, considering the razor-thin revenue improvement typically reported in the NRM literature. We also show that the policy is near-optimal, with an average gap of only 1.8% from an upper bound. More importantly, we show that the product-based bid-price policy is competitive to the dynamic programming decomposition heuristic (Liu and van Ryzin 2008, Zhang and Adelman 2009), which is considered the strongest heuristic for the NRM problem (Ma et al. 2020). In particular, the product-based bid-price policy produces a revenue lift of 0.48% on a set of hub-and-spoke instances.

The product-based bid-price policy utilizes the value of product-resource pairs, which is not obtainable through the resource-based SPL approximation. The coefficients of the product-based SPL approximation can be interpreted as the revenue contribution of each product to the value of each unit of each resource in a given period. Therefore, the product-based SPL approximation can be viewed as a generalization of the resource-based SPL approximation. Indeed, the resource-based SPL approximation can be recovered from the product-based SPL approximation.

ALPs are known to pose computational challenges due to their size. In the literature, specialized algorithms such as column generation (Adelman 2007, Meissner and Strauss 2012) and constraint sampling (de Farias and Van Roy 2004, Farias and Van Roy 2007) are proposed. However, these specialized algorithms are often very time consuming, even for moderately sized problems. For example, Meissner and Strauss (2012) show that it can take more than 10 hours to solve a small NRM problem with the resource-based SPL approximation. Tong and Topaloglu (2014) establish a compact reformulation for the affine ALPs for NRM. Subsequently, Vossen and Zhang (2015b) show that the ALPs with the resource-based SPL approximation for NRM admit compact, equivalent linear programming formulations, which they call reduced programs. These reduced programs can be solved orders of magnitude faster than the original formulations. We show that the product-based SPL approximation also has a corresponding reduced program, which substantially alleviates the computational challenge. Moreover, our computational results demonstrate that the reduced program of the product-based SPL approximation has more robust computational performance than the resource-based one; it can be solved faster and has better numerical stability.

The product-based SPL approximation also allows us to derive a set of valid inequalities, which we call *product-value inequalities*, to further tighten the reduced program. These product-value inequalities build on the observation that the difference of a product's bid-prices in two consecutive periods is bounded, as there is no more than one customer request for the product in each period.

In our study, adding product-value inequalities significantly changes policy behavior, and is therefore critical to policy performance. Product-value inequalities improve the expected revenue by approximately 0.68% on average. The maximum gain can be over 5%. In terms of computational performance, adding product-value inequalities can speed up the computation by about 25%. Taken together, our best implementation of the product-based SPL approximation is 1.42 times faster than that of the resource-based SPL approximation. Furthermore, we can achieve better numerical stability by solving more instances optimally.

Adding valid inequalities to speed up the computation of large-scale linear optimization problems is a well-known idea in the relevant literature (Adelman 2007, Vossen and Zhang 2015a). Our numerical results demonstrate that, in addition to the computational benefits, the valid inequalities also help us choose better optimal solutions for policy construction. Adding additional constraints (inequalities) is used as a way to derive relaxations of ALPs and reduce distortions in the solutions of ALPs in the literature (Nadarajah et al. 2015, Nadarajah and Secomandi 2020). One difference is that the inequalities in Nadarajah et al. (2015) and Nadarajah and Secomandi (2020), while well-motivated, are not necessarily valid in the sense that, after adding them to ALPs, the resulting relaxations of ALPs may not produce valid upper bounds.

The remainder of the paper is organized as follows. Section 2 formulates the NRM problem, reviews the resource-based SPL approximation, and illustrates the non-optimality of the resource-based bid-price policy using an example from Talluri and van Ryzin (1998). Section 3 presents the product-based SPL approximation. Section 4 reports the computational results, and Section 5 concludes. All technical proofs are in Appendix EC.1. Appendices EC.2–EC.3 report detailed numerical results.

## 2. Preliminaries: The ALP Approach for Network Revenue Management

This section introduces the dynamic programming formulation and the ALP approach for the NRM problem. We illustrate the non-optimality of the ALP using an example from Talluri and van Ryzin (1998).

There is a set of resources given by $\mathcal{I} = \{1, \ldots, I\}$, where $I$ is the total number of resources. We reserve index $i$ for resources. The resource capacity is given by the vector $\mathbf{c} = (c_1, \ldots, c_I)$, where $c_i$ is the capacity for resource $i \in \mathcal{I}$. We reserve $k$ as the index for the capacity levels. The set of products is denoted by $\mathcal{J} = \{1, \ldots, J\}$, where $J$ is the total number of products. The price of product $j$ is $f_j$. We save $j$ and $m$ as indices for the products. Let $A = [a_{ij}]$ denote an $I \times J$ matrix, where $a_{ij} \in \{0, 1\}$ is the amount of resource $i$ consumed by product $j$. We use $\mathbf{a}^j$ and $\mathbf{a}_i$ to denote the column and row of matrix $A$, respectively. With a slight abuse of notation, we also use $\mathbf{a}^j \subseteq \mathcal{I}$

to denote the set of resources used by product $j$ and use $\mathbf{a}_i \subseteq \mathcal{J}$ to denote the set of products using resource $i$.

There are $T$ discrete periods in the selling horizon. Time counts forward, so $t = 1$ is the first period. At time $T+1$, unused resources perish and there is no salvage value. We reserve $t$ as the index for the periods. As is standard in the NRM literature, we assume that the period is "small" such that there is at most one customer arrival in each period. Each arriving customer requests a specific product. A customer that requests product $j$ is called a class-$j$ customer. We assume that the probability of a class-$j$ customer arriving in period $t$ is $\lambda_{t,j}$, with $\sum_{j\in\mathcal{J}} \lambda_{t,j} \leq 1$. The probability of no customer arrival is $1 - \sum_{j\in\mathcal{J}} \lambda_{t,j}$. We focus on the independent demand model here because it is relatively simple to estimate and often provides satisfactory performance in practice (Van Ryzin 2005, Gallego et al. 2019). Moreover, it is a steppingstone for considering more complicated discrete choice models in the future.

The state of the system is given by $\mathbf{x} = (x_1, \ldots, x_I)$, where $x_i$ is the remaining capacity for resource $i$. The state space in period $t$ is then given by

$$\mathcal{X}_t = \begin{cases} \{\mathbf{c}\}, & \text{if } t = 1, \\ \left\{\mathbf{x} \in \mathbb{Z}_+^I : \mathbf{x} \leq \mathbf{c}\right\}, & \text{if } t \geq 2. \end{cases}$$

The objective is to maximize the total expected revenue by deciding which product request to accept given the state in each period $t$. The decision can be denoted by a $J$-vector $\mathbf{u}_t$, where $u_{t,j} \in \{0,1\}$ denotes whether product $j$ is accepted or not in period $t$. The decision vector $\mathbf{u}_t$ must satisfy the resource constraint; i.e., a product can only be accepted if there are sufficient resources available. The action space in state $\mathbf{x}$ is given by

$$\mathcal{U}(\mathbf{x}) = \left\{\mathbf{u}_t \in \{0,1\}^J : \mathbf{a}^j u_{t,j} \leq \mathbf{x},\ \forall j \in \mathcal{J}\right\}.$$

Let $v_t(\mathbf{x})$ be the value function, which is the maximum expected revenue from period $t$ onward given state $\mathbf{x}$ at the beginning of period $t$. The optimality equations are given by

$$v_t(\mathbf{x}) = \max_{\mathbf{u}_t \in \mathcal{U}(\mathbf{x})} \sum_j \lambda_{t,j} u_{t,j} \left(f_j + v_{t+1}(\mathbf{x} - \mathbf{a}^j)\right) + \left(1 - \sum_j \lambda_{t,j} u_{t,j}\right) v_{t+1}(\mathbf{x}), \quad \forall t, \mathbf{x}. \tag{1}$$

The boundary conditions are $v_{T+1}(\mathbf{x}) = 0$ for all $\mathbf{x} \in \mathcal{X}_{T+1}$.

It can be shown that an optimal policy accepts a booking request as long as there are sufficient resources and the price exceeds the marginal cost. That is, it is optimal to take

$$u_{t,j} = \begin{cases} \prod_{i \in \mathbf{a}^j} \mathbb{1}(x_i \geq 1), & \text{if } f_j \geq v_{t+1}(\mathbf{x}) - v_{t+1}(\mathbf{x} - \mathbf{a}^j), \\ 0, & \text{otherwise}, \end{cases} \qquad \forall t, \mathbf{x}, j. \tag{2}$$

The dynamic program (1) can be equivalently written as a linear program as follows (Adelman 2007):

$$
\begin{aligned}
\text{(LP)} \quad & \min_{\{v_t(\cdot)\}_{\forall t}} \quad v_1(\mathbf{c}) \\
& \text{s.t.} \quad v_t(\mathbf{x}) - v_{t+1}(\mathbf{x}) + \sum_j \lambda_{t,j} u_{t,j} \left( v_{t+1}(\mathbf{x}) - v_{t+1}(\mathbf{x} - \mathbf{a}^j) \right) \geq \sum_j \lambda_{t,j} u_{t,j} f_j, \\
& \qquad\qquad \forall t, \mathbf{x} \in \mathcal{X}_t, \mathbf{u}_t \in \mathcal{U}(\mathbf{x}). \quad (3)
\end{aligned}
$$

However, the number of variables and constraints in (LP) increases exponentially in the number of resources $I$ and the number of products $J$. Thus, solving (LP) is as difficult as solving the dynamic program (1). To achieve tractability, a key idea in the ALP approach is to represent the value function $v_t(\mathbf{x})$ by the weighted basis functions:

$$
v_t(\mathbf{x}) \approx \theta_t + \sum_{b \in \mathcal{B}} V_{t,b} \phi_b(\mathbf{x}), \quad \forall t, \mathbf{x} \in \mathcal{X}_t, \tag{4}
$$

where $\phi_b : \mathcal{X} \to \mathbb{R}$ for $b \in \mathcal{B}$ is a set of prespecified basis functions and $\mathcal{B}$ is some index set. The parameter $V_{t,b}$ is the weight of the basis function $\phi_b(\cdot)$ in period $t$, and $\theta_t$ is a constant offset.

Two common approximation architectures for the NRM problem are affine and SPL approximations. In this paper, we focus on the SPL approximation, which is known to be stronger than the affine approximation. The SPL approximation is given by (Farias and Van Roy 2007, Meissner and Strauss 2012, Vossen and Zhang 2015b):

$$
v_t(\mathbf{x}) \approx \theta_t + \sum_i \sum_{k=1}^{x_i} V_{t,i,k}, \quad \forall t, \mathbf{x} \in \mathcal{X}_t. \tag{5}
$$

In (5), $V_{t,i,k}$ can be interpreted as the value of the $k$-th unit of resource $i$ in period $t$. To differentiate from the approximation architecture that we propose later in the paper, we call the approximation (5) the *resource-based* SPL approximation.

Substituting (5) into (LP) yields a problem with a polynomial number of variables. However, it still has exponentially many constraints. Specialized column generation (Adelman 2007, Meissner and Strauss 2012) and constraint sampling (de Farias and Van Roy 2004, Farias and Van Roy 2007) methods were developed to handle these constraints. More recently, Vossen and Zhang (2015b) show that the ALP associated with the approximation in (5) can be written compactly as the following *reduced program*:

$$
\text{(R)} \ z_R = \max_{\mathbf{p},\mathbf{q},\mathbf{z}} \sum_{t,j} \lambda_{t,j} f_j q_{t,j}
$$

$$
\text{s.t.} \quad p_{t,i,k} = \begin{cases} 1, & \text{if } t = 1, \\ p_{t-1,i,k} \\ \quad - \sum_{j \in \mathbf{a}_i} \lambda_{t-1,j} (z_{t-1,i,j,k} - z_{t-1,i,j,k+1}), & \text{if } t > 1, \end{cases} \quad \forall t, i, k, \tag{6}
$$

$$
q_{t,j} = z_{t,i,j,1}, \quad \forall t, i, j : a_{i,j} = 1, \tag{7}
$$

$$
z_{t,i,j,k+1} \leq z_{t,i,j,k}, \quad \forall t, i, j, k : a_{i,j} = 1, \tag{8}
$$

$$
z_{t,i,j,k} \leq p_{t,i,k}, \quad \forall t, i, j, k : a_{i,j} = 1. \tag{9}
$$

The dual solution to (R) can be used to construct a bid-price control policy. According to the definition in Talluri and van Ryzin (1998), a bid-price policy specifies a set of bid-prices for each resource at each point in time, such that the request for a product is accepted if and only if there is available capacity and the price exceeds the sum of the bid-prices for all units of the resources used by the product. It is natural to use the optimal dual values of constraint (6), $\{V^*_{t,i,k}\}_{\forall t,i,k}$, to approximate the value function following (5) as follows:

$$v_t(\mathbf{x}) \approx v_t^R(\mathbf{x}) = \theta_t + \sum_i \sum_{k=1}^{x_i} V^*_{t,i,k}, \quad \forall t, \mathbf{x} \in \mathcal{X}_t. \tag{10}$$

We call (10) the *resource-based value function approximation.* A bid-price policy can be constructed such that

$$u_{t,j} = \begin{cases} \prod_{i \in \mathbf{a}^j} \mathbb{1}(x_i \geq 1), & \text{if } f_j \geq \sum\limits_{i \in \mathbf{a}^j} V^*_{t+1,i,x_i}, \\ 0, & \text{otherwise}, \end{cases} \qquad \forall t, \mathbf{x}, j. \tag{11}$$

We refer to (11) as the *resource-based* bid-price policy.

Talluri and van Ryzin (1998) use an example to illustrate the non-optimality of a particular form of bid-price policies. Their example can also be used to show that the bid-price policy specified in (11) is not optimal. We replicate their example below but change the time index to match our notation.

Example 1 (Talluri and van Ryzin (1998), Section 3.1). Consider a network with two resources and three products. There are two local products (P1 and P2), each with a price of \$250, and one through product (P3) with a price of \$500. Each local product uses one resource, while the through product uses both resources. Each resource has one unit of capacity. The problem data are shown in Table 1.

| Period ($t$) | Product: $\mathbf{a}^j$ | Price | Probability |
|---|---|---|---|
| 1 | P1: (1, 0) | \$250 | 0.3 |
| | P2: (0, 1) | \$250 | 0.3 |
| | P3: (1, 1) | \$500 | 0.4 |
| 2 | No arrival | | 0.2 |
| | P3: (1, 1) | \$500 | 0.8 |

Table 1: Problem Data for Example 1.

In period 1, the arrival probability for each local product is 0.3 and is 0.4 for the through product. In period 2, there is no demand for the local products, and the arrival probability for the through product is 0.8. An optimal policy will reject both local products and only accept the through product in period 1. If the through product does not arrive in period 1, the policy will accept the through product in period 2 if it arrives. The optimal expected revenue is \$440.

To implement the optimal policy in Example 1, the resource-based bid-prices need to satisfy $V_{2,1,1} > 250$, $V_{2,2,1} > 250$, and $V_{2,1,1} + V_{2,2,1} \leq 500$, which obviously is impossible. The best a resource-based bid-price policy can do in this example is to have $V_{2,1,1} > 250$, $V_{2,2,1} > 250$, and $V_{2,1,1} + V_{2,2,1} > 500$. Thus, the resulting policy rejects all demand in period 1 and accepts only the through itinerary (if it arrives) in period 2, yielding an expected revenue of \$400. The bad news is that even this policy is not achievable by any dual optimal solution of (R) (and the resource-based SPL approximation) because it can be verified that for this example, the maximum objective value of (R), $z_R$, is 470, which is strictly smaller than 500, and none of (R)'s dual optimal solutions can satisfy these inequalities. The best policy formed by a dual optimal solution of (R) yields an expected revenue of \$350 by accepting all demand in period 1. In the next section, we show that a new approximation can overcome the pitfall of the resource-based SPL approximation in this example.

## 3. A Product-Based Approximation for Network Revenue Management

This section proposes a product-based ALP for NRM. Section 3.1 presents a variant of the bid-price policy to avoid the pitfall mentioned at the end of Section 2. To facilitate this policy, we propose a new functional approximation for the NRM problem. We also discuss the strength of this approximation. Section 3.2 shows that the resulting ALP admits a compact reformulation (a reduced program), similar to the resource-based SPL approximation. Section 3.3 provides a set of valid inequalities for the ALP. Our numerical experiments later in the paper show that these valid inequalities improve both the policy and computational performance.

### 3.1. A Policy Taking the Network Effect into Account

The policy (11) is not optimal because it is based on the approximation (5). In this section, we propose a variant of (11) to alleviate the deviation from optimality.

The key motivation for the new policy is that the resource-based bid-price policy ignores the network effect among products sharing the same resources. Consider two products $j$ and $m$, which share resource $i$ but have different resource requirements; i.e., $i \in \mathbf{a}^j \cap \mathbf{a}^m$ and $\mathbf{a}^j \neq \mathbf{a}^m$. Suppose the remaining capacity of resource $i$ is one unit. According to the definition in Talluri and van Ryzin (1998), whether a product is accepted is completely determined by the bid-prices of the resources consumed by that product but is independent of the bid-prices of the other resources. Because resource $i$ only has one unit remaining, accepting product $j$ would prevent us from accepting product $m$ in the future. This notion suggests that the impact on product $m$ should be taken into account when deciding whether to accept a request for product $j$. How this can be achieved, however, is unclear under the resource-based SPL approximation.

As a potential remedy, we propose the following *product-based SPL approximation*:

$$v_t(\mathbf{x}) \approx \theta_t + \sum_i \sum_{j\in \mathbf{a}_i} \sum_{k=1}^{x_i} W_{t,i,j,k}, \quad \forall t, \mathbf{x} \in \mathcal{X}_t. \tag{12}$$

The term $W_{t,i,j,k}$ can be interpreted as product $j$'s revenue contribution to the value of the $k$-th unit of resource $i$ in period $t$. The product-based SPL approximation provides more granular information than the resource-based SPL approximation, given that the latter can be recovered from the former as follows:

$$V_{t,i,k} = \sum_{j\in \mathbf{a}_i} W_{t,i,j,k}, \quad \forall t, i, k. \tag{13}$$

The product-based SPL approximation provides greater flexibility for constructing policies as well. Clearly, we can construct the resource-based bid-price policy (11) via (13). More importantly, we can interpret $\sum_{i\in\mathbf{a}^j}\sum_{k=1}^{x_i} W_{t,i,j,k}$ as the total expected revenue of product $j \in \mathcal{J}$ from period $t$ onward. Therefore, given a solution $W^*$, the value function can be approximated as

$$v_t(\mathbf{x}) \approx v_t^P(\mathbf{x}) = \theta_t + \sum_j \mathbb{1}\left(\mathbf{x} \geq \mathbf{a}^j\right) \sum_{i\in\mathbf{a}^j} \sum_{k=1}^{x_i} W^*_{t,i,j,k}, \quad \forall t, \mathbf{x} \in \mathcal{X}_t. \tag{14}$$

We call (14) the *product-based value function approximation*. A salient feature of (14) is that it allows us to incorporate the capacity constraint into the approximate value function. In particular, the indicator function tracks the capacity of each product's required resources and sets the expected future revenues of products with insufficient resources to zero.

We can construct a policy by using (14) in (2) as follows. Consider two products $j, m \in \mathcal{J}$. Let $\underline{x}^{m,j} = \min\limits_{i'\in\mathbf{a}^m\cap\mathbf{a}^j} x_{i'}$. As a convention, we take $\underline{x}^{m,j} = 0$ if $\mathbf{a}^m \cap \mathbf{a}^j = \emptyset$. Thus, $\underline{x}^{m,j}$ gives the minimum capacity of the resources shared by products $j$ and $m$; it takes the value of 0 if products $j$ and $m$ do not share any resources. The bid-price that can be used to decide whether to accept product $j$ is then given by

$$\begin{aligned}
&\Delta v^P_{t+1,j}(\mathbf{x}) = v^P_{t+1,j}(\mathbf{x}) - v^P_{t+1,j}(\mathbf{x}-\mathbf{a}^j)\\
&= \theta_{t+1} + \sum_m \mathbb{1}\left(\mathbf{x} \geq \mathbf{a}^m\right) \sum_{i\in\mathbf{a}^m} \sum_{k=1}^{x_i} W^*_{t+1,i,m,k} - \left(\theta_{t+1} + \sum_m \mathbb{1}\left(\mathbf{x}-\mathbf{a}^j \geq \mathbf{a}^m\right) \sum_{i\in\mathbf{a}^m} \sum_{k=1}^{x_i - a_{i,j}} W^*_{t+1,i,m,k}\right)\\
&= \sum_{m:\mathbf{a}^m\cap\mathbf{a}^j\neq\emptyset} \left\{ \mathbb{1}\left(\mathbf{x} \geq \mathbf{a}^m\right) \sum_{i\in\mathbf{a}^m} \sum_{k=1}^{x_i} W^*_{t+1,i,m,k} - \mathbb{1}\left(\mathbf{x}-\mathbf{a}^j \geq \mathbf{a}^m\right) \sum_{i\in\mathbf{a}^m} \sum_{k=1}^{x_i - a_{i,j}} W^*_{t+1,i,m,k} \right\}\\
&= \sum_{m:\mathbf{a}^m\cap\mathbf{a}^j\neq\emptyset} \left\{ \mathbb{1}\left(\underline{x}^{m,j} = 1\right) \sum_{i\in\mathbf{a}^m} \sum_{k=1}^{x_i} W^*_{t+1,i,m,k} + \mathbb{1}\left(\underline{x}^{m,j} > 1\right) \sum_{i\in\mathbf{a}^m\cap\mathbf{a}^j} W^*_{t+1,i,m,x_i} \right\}.
\end{aligned} \tag{15}$$

The second equality holds because $\mathbb{1}\left(\mathbf{x} \geq \mathbf{a}^m\right) = \mathbb{1}\left(\mathbf{x}-\mathbf{a}^j \geq \mathbf{a}^m\right)$ when products $m$ and $j$ do not share any resources. The last equality follows by considering three cases: (i) If $\underline{x}^{m,j} > 1$, $\mathbb{1}\left(\mathbf{x} \geq \mathbf{a}^m\right) = 1$ and $\mathbb{1}\left(\mathbf{x}-\mathbf{a}^j \geq \mathbf{a}^m\right) = 1$; (ii) If $\underline{x}^{m,j} = 1$, $\mathbb{1}\left(\mathbf{x} \geq \mathbf{a}^m\right) = 1$ and $\mathbb{1}\left(\mathbf{x}-\mathbf{a}^j \geq \mathbf{a}^m\right) = 0$; (iii) If $\underline{x}^{m,j} = 0$, $\mathbb{1}\left(\mathbf{x} \geq \mathbf{a}^m\right) = 0$ and $\mathbb{1}\left(\mathbf{x}-\mathbf{a}^j \geq \mathbf{a}^m\right) = 0$.

The corresponding bid-price policy is given by

$$u_{t,j} = \begin{cases} \prod_{i \in \mathbf{a}^j} \mathbb{1}(x_i \geq 1), & \text{if } f_j \geq \Delta v^P_{t+1,j}(\mathbf{x}), \\ 0, & \text{otherwise}, \end{cases} \qquad \forall t, \mathbf{x}, j. \tag{16}$$

We call this policy the *product-based* bid-price policy. As a side note, we point out that the policy does not comply with the definition of the bid-price policies in Talluri and van Ryzin (1998). Based on this policy, when a request of product $j$ arrives, we consider all products that share resources with product $j$ to decide whether to accept the request. Let $m$ be a product that shares at least one resource with product $j$. If any of the shared resources for products $j$ and $m$ have only one unit left, we will not be able to accept future requests for product $m$ if the request for product $j$ is accepted. In this case, we add the expected future revenue of product $m$, $\sum\limits_{i \in \mathbf{a}^m} \sum_{k=1}^{x_i} W^*_{t+1,i,m,k}$, to the bid-price. However, if all shared resources for products $j$ and $m$ have capacities that are strictly higher than 1, then we only include $\sum\limits_{i \in \mathbf{a}^m \cap \mathbf{a}^j} W^*_{t+1,i,m,x_i}$ in the bid-price.

To emphasize the difference between the product- and resource-based bid-prices, we rewrite the product-based bid-prices (15) as follows:

$$\begin{aligned} &\sum_{m: \mathbf{a}^m \cap \mathbf{a}^j \neq \emptyset} \left\{ \mathbb{1}\left(\underline{x}^{m,j} = 1\right) \sum_{i \in \mathbf{a}^m} \sum_{k=1}^{x_i} W^*_{t+1,i,m,k} + \mathbb{1}\left(\underline{x}^{m,j} > 1\right) \sum_{i \in \mathbf{a}^m \cap \mathbf{a}^j} W^*_{t+1,i,m,x_i} \right\} \\ = &\sum_{i \in \mathbf{a}^j} \sum_{m \in \mathbf{a}_i} W^*_{t+1,i,m,x_i} + \sum_{m: \mathbf{a}^m \cap \mathbf{a}^j \neq \emptyset} \left\{ \mathbb{1}\left(\underline{x}^{m,j} = 1\right) \left[ \sum_{i \in \mathbf{a}^m \cap \mathbf{a}^j} \sum_{k=1}^{x_i - 1} W^*_{t+1,i,m,k} + \sum_{i \in \mathbf{a}^m \setminus \mathbf{a}^j} \sum_{k=1}^{x_i} W^*_{t+1,i,m,k} \right] \right\}. \end{aligned}$$

The first term, $\sum_{i \in \mathbf{a}^j} \sum_{m \in \mathbf{a}_i} W^*_{t+1,i,m,x_i}$, is the same as the resource-based bid-price policy based on (11) via (13). Therefore, the product-based bid-price policy has a higher bid-price than the resource-based bid-price policy when any shared resource has only one unit of capacity left. It is well known that the main trade-off in the NRM problem is to strike a balance between the immediate revenue from accepting a product request and the revenue from potentially more profitable product requests in the future. The product-based bid-price policy uses more stringent bid-prices to protect the capacity for potentially more profitable future product requests when the capacity is low.

In Example 1, the pitfall mentioned earlier is no longer a problem if we use the product-based bid-price policy. An optimal solution based on the product-based SPL approximation is given by $W^*_{1,1,1,1} = 215$, $W^*_{1,2,2,1} = 255$, $W^*_{2,1,3,1} = 200$, $W^*_{2,2,3,1} = 200$, and 0's for the other variables. According to (14), the value functions are approximated as: $v^P_2(1,1) = 400$, $v^P_2(1,0) = v^P_2(0,1) = v^P_2(0,0) = 0$. In period 1, both local products are rejected because $f_1 = 250 < v^P_2(1,1) - v^P_2(1,0) = 400$ and $f_2 = 250 < v^P_2(1,1) - v^P_2(0,1) = 400$. However, the through product is accepted as $f_3 = 500 > v^P_2(1,1) - v^P_2(0,0) = 400$. Then, in period 2, when there is enough capacity, an arriving request for the through product is accepted because $f_3 = 500 > 0$, given that $W_{3,i,j,k} = 0$ for all $i, j, k$. This policy yields the optimal expected revenue of \$440.

Given that we use the product-based value function approximation (14) to construct policies, it is natural to consider solving the ALP corresponding to this approximation. That is, we may consider the following value function approximation:

$$v_t(\mathbf{x}) \approx \theta_t + \sum_j \mathbb{1}\left(\mathbf{x} \geq \mathbf{a}^j\right) \sum_{i \in \mathbf{a}^j} \sum_{k=1}^{x_i} \underline{W}_{t,i,j,k}, \quad \forall t, \mathbf{x} \in \mathcal{X}_t. \tag{17}$$

The difficulty is that the ALP corresponding to the approximation (17), denoted by ($\mathrm{D}^{\underline{W}}$) shown in Appendix EC.1.1, is computationally challenging. However, we show that it has an appealing theoretical property: its approximation error is smaller than the resource-based SPL approximation.

Given an optimal solution $(\theta^*, \underline{W}^*)$ to ($\mathrm{D}^{\underline{W}}$), we can approximate the value functions as:

$$v_t(\mathbf{x}) \approx v_t^{\underline{W}}(\mathbf{x}) = \theta_t^* + \sum_j \mathbb{1}\left(\mathbf{x} \geq \mathbf{a}^j\right) \sum_{i \in \mathbf{a}^j} \sum_{k=1}^{x_i} \underline{W}^*_{t,i,j,k}, \quad \forall t, \mathbf{x} \in \mathcal{X}_t. \tag{18}$$

Theorem 1. *$v_t^{\underline{W}}(\mathbf{x})$ has a smaller approximation error than the resource-based value function approximation $v_t^R(\mathbf{x})$ for all $t$, $\mathbf{x} \in \mathcal{X}_t$.*

While Theorem 1 provides an interesting theoretical result, it is not computationally viable to solve ($\mathrm{D}^{\underline{W}}$) to optimality. ($\mathrm{D}^{\underline{W}}$) is a linear program with exponentially many constraints. The pricing problem of its dual (or its own constraint separation problem) does not satisfy the conditions in Lemma 2 in Vossen and Zhang (2015b). Thus, ($\mathrm{D}^{\underline{W}}$) does not have an equivalent compact reformulation. We must resort to specialized algorithms such as column generation (Adelman 2007, Meissner and Strauss 2012) or constraint sampling (de Farias and Van Roy 2004, Farias and Van Roy 2007) methods, which impose significant computational burdens (see Meissner and Strauss 2012). To achieve computational efficiency, we solve the product-based SPL approximation, which admits a compact reformulation (P) presented in Section 3.2, and construct a policy using (14). As evident later in our computational studies, this strategy produces a fruitful outcome, as it strikes a balance between computational tractability and a reduction in approximation errors.

### 3.2. The ALP Based on the Product-Based SPL Approximation

This section shows that the ALP based on the product-based SPL approximation admits an equivalent compact reformulation, which we call the reduced program. This result echoes a similar one for the resource-based SPL approximation shown in Vossen and Zhang (2015b).

Plugging (12) into (LP) and simplifying the constraints, we obtain

$$\begin{aligned}
(\mathrm{D}') \min_{\theta, W} \quad & \theta_1 + \sum_j \sum_{i \in \mathbf{a}^j} \sum_{k=1}^{c_i} W_{1,i,j,k} \\
\text{s.t.} \quad & \theta_t - \theta_{t+1} + \sum_j \sum_{i \in \mathbf{a}^j} \sum_{k=1}^{x_i} (W_{t,i,j,k} - W_{t+1,i,j,k}) \\
& + \sum_j \lambda_{t,j} u_{t,j} \sum_{i \in \mathbf{a}^j} \sum_{m \in \mathbf{a}_i} W_{t+1,i,m,x_i} \geq \sum_j \lambda_{t,j} u_{t,j} f_j, \qquad \forall t, \mathbf{x} \in \mathcal{X}_t, \mathbf{u}_t \in \mathcal{U}(\mathbf{x}).
\end{aligned} \tag{19}$$

Let $\mathbf{h}$ be the dual variables of constraint (19). The corresponding dual program is

$$
\begin{aligned}
(\mathrm{P}') \quad \max_{h} \quad & \sum_{t} \sum_{\mathbf{x}\in\mathcal{X}_t, \mathbf{u}\in\mathcal{U}(\mathbf{x})} \left( \sum_{j} \lambda_{t,j} u_{t,j} f_j \right) h_{t,\mathbf{x},\mathbf{u}} \\
\text{s.t.} \quad & \sum_{\mathbf{x}\in\mathcal{X}_t, \mathbf{u}\in\mathcal{U}(\mathbf{x})} h_{t,\mathbf{x},\mathbf{u}} = 1, \qquad \forall t, \qquad (20) \\
& \sum_{\substack{\mathbf{x}\in\mathcal{X}_t, \mathbf{u}\in\mathcal{U}(\mathbf{x}) \\ : x_i \geq k}} h_{t,\mathbf{x},\mathbf{u}} = \begin{cases} 1, & \text{if } t = 1, \\ \displaystyle\sum_{\substack{\mathbf{x}\in\mathcal{X}_{t-1}, \mathbf{u}\in\mathcal{U}(\mathbf{x}) \\ : x_i \geq k}} h_{t-1,\mathbf{x},\mathbf{u}} \\ \quad - \displaystyle\sum_{\substack{\mathbf{x}\in\mathcal{X}_{t-1}, \mathbf{u}\in\mathcal{U}(\mathbf{x}) \\ : x_i = k}} \sum_{m} \lambda_{t-1,m} a_{i,m} u_{t-1,j} h_{t-1,\mathbf{x},\mathbf{u}}, & \text{if } t > 1, \end{cases} \\
& \qquad \forall t, i, j, k : a_{i,j} = 1, \qquad (21) \\
& \mathbf{h} \geq 0.
\end{aligned}
$$

Define the following variables: $q_{t,j} = \sum_{\mathbf{x}\in\mathcal{X}_t, \mathbf{u}\in\mathcal{U}(\mathbf{x})} u_{t,j} h_{t,\mathbf{x},\mathbf{u}}$, for all $t$ and $j$, $p_{t,i,k} = \sum_{\substack{\mathbf{x}\in\mathcal{X}_t, \mathbf{u}\in\mathcal{U}(\mathbf{x}) \\ : x_i \geq k}} h_{t,\mathbf{x},\mathbf{u}}$, for all $t$, $i$, and $k$, $z_{t,j,i,k} = \sum_{\substack{\mathbf{x}\in\mathcal{X}_t, \mathbf{u}\in\mathcal{U}(\mathbf{x}) \\ : x_i \geq k}} u_{t,j} h_{t,\mathbf{x},\mathbf{u}}$, for all $t$, $j$, $i$, and $k$ such that $a_{ij} = 1$. Using these new variables in (P′), we have

$$
\begin{aligned}
(\mathrm{P}) \; z_P = \max_{\mathbf{p},\mathbf{q},\mathbf{z}} \; & \sum_{t,j} \lambda_{t,j} f_j q_{t,j} \\
\text{s.t.} \quad & p_{t,i,k} = \begin{cases} 1, & \text{if } t = 1, \\ p_{t-1,i,k} \\ \quad - \displaystyle\sum_{m\in\mathbf{a}_i} \lambda_{t-1,m} (z_{t-1,i,m,k} - z_{t-1,i,m,k+1}), & \text{if } t > 1, \end{cases} \qquad \forall t, i, j, k : a_{i,j} = 1, \qquad (22) \\
& q_{t,j} = z_{t,i,j,1}, \qquad \forall t, i, j : a_{i,j} = 1, \qquad (23) \\
& z_{t,i,j,k+1} \leq z_{t,i,j,k}, \qquad \forall t, i, j, k : a_{i,j} = 1, \qquad (24) \\
& z_{t,i,j,k} \leq p_{t,i,k}, \qquad \forall t, i, j, k : a_{i,j} = 1. \qquad (25)
\end{aligned}
$$

Constraint (22) is derived directly from the second constraint of (P′). As a notational convention, we take $z_{t,i,j,c_i+1} = 0$ for all $t, j, i$. Constraints (23)–(25) are added to enforce the definition of variables $\mathbf{q}$, $\mathbf{p}$, and $\mathbf{z}$.

Theorem 2 below shows that the formulations (P) and (P′) are equivalent. To establish this result, we use a different and more direct argument compared to the one in Vossen and Zhang (2015b), which relies on the structure of the ALP's column generation subproblems.

Theorem 2. *The formulations (P) and (P′) are equivalent in the sense that they have the same objective value.*

We show that the product-based SPL approximation provides the same bound as the resource-based SPL approximation in the following proposition. Proposition 1 establishes (P) as a reformulation of (R).

Proposition 1. *We have* $z_P = z_R \geq v_1(\mathbf{c})$, *where* $v_1(\mathbf{c})$ *is the optimal value of the dynamic programming model* (1).

### 3.3. Dual Valid Inequalities for (P)

The dual solution $W^*$ of constraint (22) can be used to construct the product-based bid-price policy. For Example 1, we have shown that the product-based bid-price policy is optimal given an optimal dual solution. However, when the optimal dual solution is not unique, not all optimal dual solutions can achieve this desired result.

To illustrate the issue with multiple optimal solutions, we consider Example 1 again. A dual optimal solution of (P) is $W^*_{1,1,1,1} = 215$, $W^*_{1,2,2,1} = 255$, $W^*_{2,1,1,1} = 200$, $W^*_{2,2,2,1} = 200$ and 0's for the other variables. Then, the approximate value functions are as follows: $v_2^P(1,1) = 400$, $v_2^P(1,0) = 200$, $v_2^P(0,1) = 200$, and $v_2^P(0,0) = 0$. The policy is $u_{1,1} = 1$, given $f_1 > v_2^P(1,1) - v_2^P(0,1) = 200$, $u_{1,2} = 1$, given $f_2 > v_2^P(1,1) - v_2^P(1,0) = 200$, $u_{1,3} = 1$, given $f_3 > v_2^P(1,1) - v_2^P(0,0) = 400$ and $u_{2,3} = 1$. The corresponding expected revenue is \$350, which is suboptimal. The issue here is that this dual optimal solution does not correctly reflect the expected revenue contribution from each product. In period 2, the expected revenue from product 1 is 0, while $W^*_{2,1,1,1} = 200$. Similarly, the expected revenue from product 2 is 0, while $W^*_{2,2,2,1} = 200$. Ideally, we should have $W^*_{2,1,1,1} = W^*_{2,2,2,1} = 0$ in a dual optimal solution of (P).

The two aforementioned dual optimal solutions of (P) for Example 1 actually correspond to the same dual optimal solution of (R) as $V^*_{1,1,1} = 215$, $V^*_{1,2,1} = 255$, $V^*_{2,1,1} = 200$, and $V^*_{2,2,1} = 200$. Because of (13), each dual optimal solution of (P) corresponds to one dual optimal solution of (R). Therefore, the product-based approximation can be viewed as a way to disaggregate a dual optimal solution of (R). However, there could be multiple ways to do such a disaggregation. Some of them cannot take advantage of the product-based bid-price policy because they cannot correctly capture the expected revenue contribution from the products (e.g., the one in the last paragraph).

One way to address this issue is to add a set of valid inequalities to the dual of (P). These valid inequalities help us select an optimal solution with the potential to produce a more effective product-based bid-price policy. Furthermore, although Theorem 1 offers the hope that our approach leads to a stronger policy, it is not guaranteed. However, we may add valid inequalities to impose additional structures and reduce distortions in the optimal solution. This is similar to the inequalities for reducing distortions in Nadarajah et al. (2015) and Nadarajah and Secomandi (2020).

Another benefit of adding these inequalities is that it speeds up the convergence of the LP solution. Adding dual-optimal inequalities to accelerate and stabilize LP solutions is a well-known technique in the mathematical programming literature; see, e.g., Ben Amor et al. (2006) and Gschwind and Irnich (2016). In the literature on the LP-based ADP for NRM, Adelman (2007) and Vossen and Zhang (2015a) also use valid inequalities to speed up the computation. Our numerical results in Section 4 show that adding these inequalities indeed improves the computational performance and is critical for improving the policy performance.

Next, we derive a set of valid inequalities for the dual of (P). Let $(W, \beta, \gamma, \delta)$ be the dual variables associated with constraints (22)–(25). The dual of (P) is given by

$$
\begin{aligned}
\text{(D)} \quad \min_{W,\beta,\gamma,\delta} \quad & \sum_{j} \sum_{i \in \mathbf{a}^j} \sum_{k=1}^{c_i} W_{1,i,j,k} \\
\text{s.t.} \quad & \sum_{j \in \mathbf{a}_i} W_{t,i,j,k} - \sum_{j \in \mathbf{a}_i} W_{t+1,i,j,k} = \sum_{j \in \mathbf{a}_i} \delta_{t,i,j,k}, && \forall t, i, k, && (26) \\
& \sum_{i \in \mathbf{a}^j} \beta_{t,i,j} = \lambda_{t,j} f_j, && \forall t, j, && (27) \\
& \delta_{t,i,j,k} - \gamma_{t,i,j,k} + \lambda_{t,j} \sum_{m \in \mathbf{a}_i} W_{t+1,i,m,k} \\
& \qquad = \begin{cases} \beta_{t,i,j}, & \text{if } k = 1, \\ \lambda_{t,j} \sum\limits_{m \in \mathbf{a}_i} W_{t+1,i,m,k-1} - \gamma_{t,i,j,k-1}, & \text{if } k \geq 2, \end{cases} && \forall t, i, j, k : a_{i,j} = 1, && (28) \\
& \gamma_{t,i,j,k}, \delta_{t,i,j,k} \geq 0, && \forall t, i, j, k : a_{i,j} = 1. && (29)
\end{aligned}
$$

We first establish some properties of the optimal solution. These properties generalize the monotonicity result in Adelman (2007) and Vossen and Zhang (2015b), allowing us to establish a set of valid inequalities.

Lemma 1. *There exists an optimal solution* $(W^*, \beta^*, \gamma^*, \delta^*)$ *to (D) where*

$$
W^*_{t,i,j,k} \geq W^*_{t+1,i,j,k} \geq 0, \qquad \forall t, i, j, k : a_{i,j} = 1, \tag{30}
$$

$$
\delta^*_{t,i,j,k} = W^*_{t,i,j,k} - W^*_{t+1,i,j,k}, \qquad \forall t, i, j, k : a_{i,j} = 1, \tag{31}
$$

$$
\sum_{j \in \mathbf{a}_i} W^*_{t,i,j,k} \geq \sum_{j \in \mathbf{a}_i} W^*_{t,i,j,k+1}, \qquad \forall t, i, k. \tag{32}
$$

We can interpret the structural properties in Lemma 1 as follows. Inequality (30) means that the expected revenue from product $j$ to the value of the $k$-th unit of resource $i$ is non-increasing over time. Furthermore, the difference in the value between two consecutive periods can be captured by the corresponding $\delta$ variable, as stated in (31). Lastly, for each resource $i$ and period $t$, (32) states that the value of the $k$-th unit is not smaller than that of the $(k+1)$-th unit. Thus, the value of an additional unit of a resource is non-increasing in a given period.

Building on Lemma 1, Proposition 2 below provides a set of valid inequalities, which are referred to as *product-value inequalities*. Recall that $W_{t,i,j,k}$ represents the revenue contribution from product $j$ to the value of the $k$-th unit of resource $i$ in period $t$. Then, for a given state $\mathbf{x}$ (the current resource level), $\sum_{i \in \mathbf{a}^j} \sum_{k=1}^{x_i} W_{t,i,j,k}$ can be interpreted as the total expected revenue of product $j$ from period $t$ onward. Intuitively, we have $\sum_{i \in \mathbf{a}^j} \sum_{k=1}^{x_i} W_{T,i,j,k} \leq \lambda_{T,j} f_j$ in the last period. The expected revenue of product $j$ in period $t < T$ is no more than $\lambda_{t,j} f_j$. Thus, $\sum_{i \in \mathbf{a}^j} \sum_{k=1}^{x_i} W_{t,i,j,k}$ is not greater than $\sum_{i \in \mathbf{a}^j} \sum_{k=1}^{x_i} W_{t+1,i,j,k} + \lambda_{t,j} f_j$. Proposition 2 formalizes these results.

PROPOSITION 2. *There exists an optimal solution to (D) that satisfies*

$$\sum_{i\in\mathbf{a}^j}\sum_{k=1}^{x_i} W_{t,i,j,k} - \sum_{i\in\mathbf{a}^j}\sum_{k=1}^{x_i} W_{t+1,i,j,k} \le \lambda_{t,j} f_j, \quad \forall t, j, \mathbf{x}\in\mathcal{X}_t. \tag{33}$$

The valid inequalities (33) hold for all $t, j, \mathbf{x}\in\mathcal{X}_t$. Enumerating these inequalities can be difficult even for very small problems because the set of state $\mathbf{x}$ can be very large. However, we show that by taking $\mathbf{x}=\mathbf{c}$ in (33), we arrive at a set of dominating inequalities:

$$\sum_{i\in\mathbf{a}^j}\sum_{k=1}^{c_i} W_{t,i,j,k} - \sum_{i\in\mathbf{a}^j}\sum_{k=1}^{c_i} W_{t+1,i,j,k} \le \lambda_{t,j} f_j, \quad \forall t, j. \tag{34}$$

To see this, note that $W$ is monotone in $t$ according to (30). Hence

$$\begin{aligned}&\sum_{i\in\mathbf{a}^j}\sum_{k=1}^{x_i} W_{t,i,j,k} - \sum_{i\in\mathbf{a}^j}\sum_{k=1}^{x_i} W_{t+1,i,j,k} = \sum_{i\in\mathbf{a}^j}\sum_{k=1}^{x_i}(W_{t,i,j,k} - W_{t+1,i,j,k})\\ &\le \sum_{i\in\mathbf{a}^j}\sum_{k=1}^{c_i}(W_{t,i,j,k} - W_{t+1,i,j,k}) = \sum_{i\in\mathbf{a}^j}\sum_{k=1}^{c_i} W_{t,i,j,k} - \sum_{i\in\mathbf{a}^j}\sum_{k=1}^{c_i} W_{t+1,i,j,k} \le \lambda_{t,j} f_j, \quad \forall t, j, \mathbf{x}\in\mathcal{X}_t.\end{aligned}$$

This result is formally stated in the following proposition.

PROPOSITION 3. *Inequalities* (34) *dominate inequalities* (33)*.*

Given Proposition 3, we can focus on inequalities (34) instead of inequalities (33). Importantly, inequalities (34) are indexed by time and product and are much more compact than inequalities (33), which grow exponentially in the number of resources.

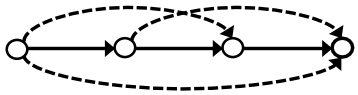
Adding (34) to (D) gives us the following formulation:

$$\begin{aligned}(\mathrm{D}_{\mathbf{c}})\quad \min_{W,\beta,\gamma,\delta}\quad & \sum_{i}\sum_{j\in\mathbf{a}_i}\sum_{k=1}^{c_i} W_{1,i,j,k}\\ \text{s.t.}\quad & (26)-(29),\\ & \sum_{i\in\mathbf{a}^j}\sum_{k=1}^{c_i} W_{t,i,j,k} - \sum_{i\in\mathbf{a}^j}\sum_{k=1}^{c_i} W_{t+1,i,j,k} \le \lambda_{t,j} f_j, \quad \forall t, j.\end{aligned}$$

Our computational studies in Section 4 confirm that enforcing (34) can improve both the computational and policy performance.

## 4. Computational Results

This section reports the results of our computational study, which has two objectives: i) to examine the performance of the product-based bid-price policies and ii) to evaluate the computational performance of the reformulations and the impact of the product-value inequalities. The computational study is performed on a MacBook Pro with the M1 Max CPU, 64 GB RAM, and macOS operating system. The computer code is implemented in Julia (Bezanson et al. 2017) with JuMP (Dunning et al. 2017). The solver is Gurobi 9.5.1 (Gurobi Optimization 2022).



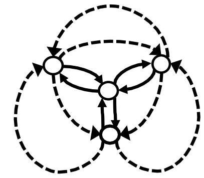


Figure 1: (a) A Hotel Instance with Three Nights. (b) A Hub-and-Spoke Instance with 3 Spokes

### 4.1. Test Instances

We consider two sets of NRM instances with different network structures: hotel instances and hub-and-spoke instances. In the hotel instances, each resource corresponds to a room-night (see, e.g., Gallego and van Ryzin 1997). There are $I$ resources in each instance. The products are requests to stay in a room for several consecutive nights. Starting at night $i = 1, 2, \cdots, I$, a customer can request to stay for $i' = 1, 2, \cdots, I - i + 1$ consecutive nights. Thus, there are $\frac{I(I+1)}{2}$ types of stay-over requests. Figure 1(a) shows an example of a hotel-network instance with three nights. The solid arrows represent the resources and one-night-stay requests. The dashed arrows represent multiple-night-stay requests. There are two classes for each itinerary, where the revenue of a high-class is $\kappa$ times that of its corresponding low-class. The revenue for the low-class for each night $i$, $f_i$, is drawn from a discrete uniform distribution between 1 and 100. The total revenue generated by a low-class request starting at night $i$ and staying for $i'$ nights is $\sum_{\bar{i}=i}^{i+i'-1} f_{\bar{i}}$. The arrival probabilities $\lambda_{t,j}$ for all $t \in \mathcal{T}$ and $j \in \mathcal{J}$ are generated using the method described in Section 5.1 of Ma et al. (2020). This method captures the phenomenon that high-class requests tend to arrive later in the selling horizon. The total expected demand for the capacity on night $i$ is $\sum_t \sum_j a_{i,j} \lambda_{t,j}$. Let $\alpha$ denote the load factor in the system. We set the initial capacity of night $i$ as $c_i = \left\lceil \frac{\sum_t \sum_j a_{i,j} \lambda_{t,j}}{\alpha} \right\rceil$. We consider hotel instances with a different selling horizon $T$, number of resources $I$, price ratio $\kappa$, and load factor $\alpha$. By taking $T \in \{50, 100, 200, 400\}$, $I \in \{2, 3, 4, 5\}$, $\kappa \in \{4, 8\}$, and $\alpha \in \{1.6, 2.2, 3.0\}$, we generate 96 hotel instances.

The hub-and-spoke instances are generated similarly, except that the underlying networks are different. Hub-and-spoke networks are based on the ones in Topaloglu (2009) and have been used in computational experiments in several other papers (Vossen and Zhang 2015a,b, Kunnumkal and Talluri 2016, Brown and Smith 2014). The resources correspond to flight legs, and the products correspond to itineraries with different prices. In each instance, there are $N$ spokes and one hub. One flight leg goes from the hub to each non-hub location, and another flight leg exists in the opposite direction. Therefore, there are $2N$ legs in total ($I = 2N$). We have $2N$ itineraries between the non-hub locations and the hub, and there are $N(N-1)$ itineraries between the non-hub locations. Thus, there are $N^2 + N$ itineraries in total. Figure 1(b) shows an example of a hub-and-spoke instance with three spokes. The solid arrows represent the resources and one-leg itineraries. The dashed arrows represent the through itineraries from one non-hub location to another. For

| | | % Gain To RB | | | | To Upp. Bnd. | | To DEC | Average Running Time (s) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Avg | | Maximum | | Avg % Gap | | Avg % | `BarTol=1e-6` | | | |
| | T | PB | $\text{PB}_\mathbf{c}$ | PB | $\text{PB}_\mathbf{c}$ | RB | $\text{PB}_\mathbf{c}$ | $\text{PB}_\mathbf{c}$ | (R) | (P) | ($\text{D}_\mathbf{c}$) | DEC |
| Hotel | 50 | 0.183 | 0.783 | 0.922 | 2.468 | 3.195 | 2.441 | -0.996 | 1.2 | 0.8 | 0.7 | 0.4 |
| | 100 | 0.175 | 0.707 | 0.552 | 3.362 | 2.523 | 1.839 | -0.911 | 5.6 | 4.6 | 3.3 | 1.5 |
| | 200 | 0.131 | 1.304 | 0.344 | 4.327 | 2.622 | 1.363 | -0.787 | 32.7 | 29.3 | 21.6 | 6.4 |
| | 400 | 0.048 | 1.643 | 0.177 | 5.270 | 2.534 | 0.952 | -0.591 | 185.5 | 153.2 | 126.5 | 25.6 |
| | All | 0.134 | 1.109 | | | 2.718 | 1.648 | -0.821 | 56.3 | 47.0 | 38.0 | 8.5 |
| Hub-Spoke | 50 | -0.112 | 0.285 | 1.179 | 1.051 | 3.220 | 2.942 | 1.571 | 0.5 | 0.4 | 0.5 | 0.9 |
| | 100 | -0.074 | 0.177 | 0.208 | 1.540 | 2.179 | 2.007 | 0.697 | 2.7 | 2.0 | 2.0 | 3.3 |
| | 200 | -0.018 | 0.325 | 0.343 | 1.401 | 1.799 | 1.480 | 0.060 | 13.3 | 11.0 | 9.7 | 13.8 |
| | 400 | -0.024 | 0.515 | 0.133 | 2.768 | 1.826 | 1.328 | -0.408 | 64.2 | 55.8 | 51.7 | 55.8 |
| | All | -0.057 | 0.326 | | | 2.256 | 1.939 | 0.480 | 20.2 | 17.3 | 16.0 | 18.5 |
| Overall | | 0.039 | 0.717 | | | 2.487 | 1.794 | -0.171 | 38.2 | 32.1 | 27.0 | 13.5 |

Table 2: Summary of the Policy Performance and Average Running Time with `BarTol=1e-6` in Seconds

each itinerary, there are two classes. There are 96 hub-and-spoke instances corresponding to $T \in \{50, 100, 200, 400\}$, $N \in \{2, 3, 4, 5\}$ (i.e., $I \in \{4, 6, 8, 10\}$), $\kappa \in \{4, 8\}$, and $\alpha \in \{1.6, 2.2, 3.0\}$. The URL `http://dx.doi.org/10.17632/tn2dzcjmkr.1` provides these 192 instances.

### 4.2. Performance of the Product-Based Bid-Price Policy

We consider four policies. The first policy, denoted by RB, is the resource-based bid-price policy based on (11). The next two are product-based bid-price policies based on (16). The second policy, denoted by PB, is based on the dual solution of (P). To have a fair comparison and isolate the effect of the product-based bid-prices, we use the same dual optimal solution of (P) to construct RB and PB. RB is constructed based on (11) via (13). The third policy, denoted by $\text{PB}_\mathbf{c}$, is based on the solution of ($\text{D}_\mathbf{c}$). The last policy we have included as a benchmark is the policy obtained from the dynamic programming decomposition (DEC) approach, which decomposes the problem by resources. The details of DEC can be found in Zhang and Adelman (Section 4.2, 2009). DEC is often considered the strongest heuristic in practice for the NRM problem, which is verified by the extensive experimental results in Ma et al. (2020). We also implemented and tested another version of DEC based on Liu and van Ryzin (Section 6.3.2, 2008). However, the performance of this variant is dominated by the one based on Zhang and Adelman (Section 4.2, 2009). Thus, we choose not to report its results. We estimate the expected revenue of each policy by simulating 100,000 sample paths for each instance. The detailed numerical results are relegated to Appendix EC.2.1 and Appendix EC.2.2 for the hotel and hub-and-spoke instances, respectively.

We provide an overview of the results in Table 2. The first column in the table specifies the network structure. The number of periods, $T$, is shown in the second column. The third column reports the average and maximum percent revenue gains of PB and $\text{PB}_\mathbf{c}$ over RB. Taking $\text{PB}_\mathbf{c}$ as an example, the percent revenue gain is calculated as ($\text{PB}_\mathbf{c}$ - RB)/RB. The fourth column shows the average percent revenue gaps of RB and $\text{PB}_\mathbf{c}$ from the upper bounds. The fifth column is

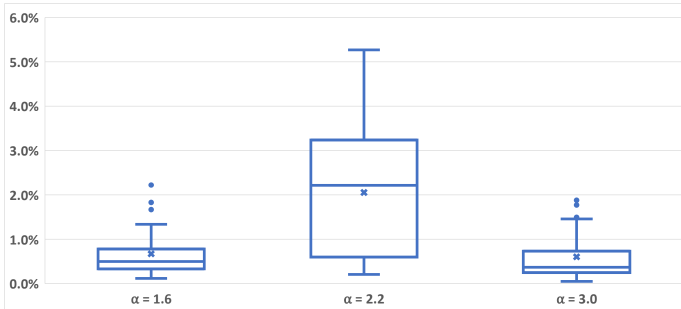



Figure 2: Box-Plot of Revenue Gains for the Hotel Instances with Various Load Factors ($\alpha$).

the average percent difference between $\text{PB}_\mathbf{c}$ and DEC, which is calculated as ($\text{PB}_\mathbf{c}$ - DEC)/DEC. We observe that product-value inequalities are critical to product-based policies. Without them, PB provides a small improvement on the hotel instances and performs worse on the hub-and-spoke instances than RB. With them, $\text{PB}_\mathbf{c}$ significantly outperforms RB on both the hotel and hub-and-spoke instances. Hence, we focus on $\text{PB}_\mathbf{c}$ hereafter.

We first examine the hotel instances. When $T = 50$, the average additional revenue gain of $\text{PB}_\mathbf{c}$ compared to RB is 0.783%. The maximum additional gain is more than 2.4%. As $T$ increases, the average additional gain of $\text{PB}_\mathbf{c}$ becomes larger. When $T = 400$, the average revenue gain is 1.643%, the maximum one is 5.270%, and the average gap from the upper bounds is only 0.952%. Across all hotel instances, the average additional gain of $\text{PB}_\mathbf{c}$ is 1.109%, which is an improvement of 0.975% over that of PB. Also, with 5% statistical significance (a p-value less than 0.05), $\text{PB}_\mathbf{c}$ achieves higher expected revenues than RB in 95 out of the 96 (about 99%) hotel instances based on the one-tailed Welch's t-test. Moreover, $\text{PB}_\mathbf{c}$ is about 1.645% away from the upper bound on average. When compared to DEC, the average revenues of $\text{PB}_\mathbf{c}$ are about 0.821% less than that of DEC for the hotel instances.

We examine different parameters and observe an interesting pattern related to the load factors. To better understand the impact of the load factors on the performance of $\text{PB}_\mathbf{c}$, we create a box-plot of revenue gains with respect to load factor $\alpha$ in Figure 2. When $\alpha = 2.2$, $\text{PB}_\mathbf{c}$ shows the highest impact. The average gains are 0.671%, 2.054%, and 0.603% for $\alpha = 1.6$, 2.2, and 3.0, respectively. We also tested instances with load factors other than these three values in our preliminary experiments. However, $\text{PB}_\mathbf{c}$ generates little gain there. This observation emphasizes that when resources are abundant ($\alpha$ is small) or scarce ($\alpha$ is big), switching from resource-based policies to product-based ones has little impact.

One surprising finding is that $\text{PB}_\mathbf{c}$ generates higher gains when the selling horizon, $T$, increases. To investigate this phenomenon, we create four plots in Figure 3 to investigate the behavior of RB, PB, and $\text{PB}_\mathbf{c}$. Each plot corresponds to a different selling horizon. Each data point is between 0 and 1 and represents the proportion of accepted product requests by a policy in that period across all

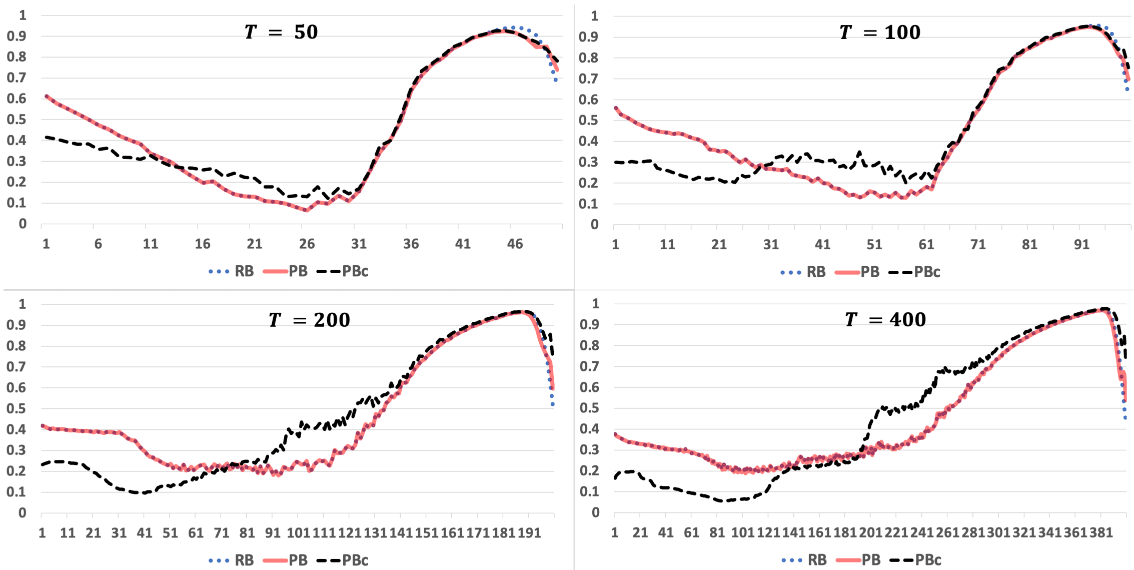



Figure 3: Accepted Proportion of Product Requests for RB, PB, and $\text{PB}_{\mathbf{c}}$ Over All Hotel Instances with $T$ Periods Based on 100,000 Samples.

instances in our simulation. RB, PB, and $\text{PB}_{\mathbf{c}}$ are shown as blue dotted, red solid, and black dashed lines, respectively. One immediate observation is that RB and PB are almost identical to each other. They only differ in the periods closer to the end of the selling horizon. In contrast, $\text{PB}_{\mathbf{c}}$ behaves quite differently compared to RB. $\text{PB}_{\mathbf{c}}$ rejects more requests in earlier periods and accepts more requests in later periods. This tendency becomes more significant as the selling horizon increases. This finding suggests that $\text{PB}_{\mathbf{c}}$ tends to use more stringent bid-prices to protect the capacity for high-class requests in the later periods of the selling horizon and demonstrates the critical role of the product-value inequalities.

Regarding the hub-and-spoke instances, the observations are quite similar to those for the hotel instances. However, product-value inequalities are even more crucial here because PB actually performs worse than RB. With 5% statistical significance, $\text{PB}_{\mathbf{c}}$ has higher expected revenues than RB in 64 out of the 96 (about 67%) instances based on Welch's t-test. Compared to RB, on average, $\text{PB}_{\mathbf{c}}$ improves the average revenues by 0.326% and reduces the gap from 2.256% to 1.939%. Importantly, it is worth noting that $\text{PB}_{\mathbf{c}}$ is stronger than DEC for most of the instances, especially when the selling horizon is shorter, and has 0.480% higher average revenue overall.

Overall, our numerical results demonstrate several points. First, the product-based bid-price policy can be better than the resource-based bid-price policy. $\text{PB}_{\mathbf{c}}$ can improve average revenues by 0.72% and has higher revenue than RB for 159 out of 192 instances at a 5% significance level. Second, product-value inequalities are extremely critical as they enable $\text{PB}_{\mathbf{c}}$ to behave in a more intelligent way for obtaining most of the gain. $\text{PB}_{\mathbf{c}}$ has higher revenue than PB for 166 out of 192 instances at a 5% significance level. Third, $\text{PB}_{\mathbf{c}}$ is competitive to DEC, which is often considered the strongest heuristic for the NRM problem in practice.

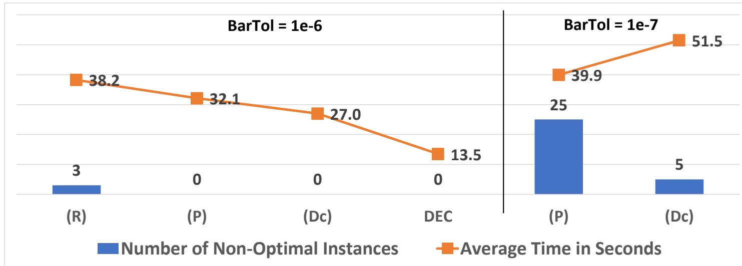



Figure 4: Summary of Computational Performance with `BarTol=1e-6` and `BarTol=1e-7`.

### 4.3. Computational Performance

This section presents the computational performance of the three formulations: (R), (P), and ($\mathrm{D}_\mathbf{c}$). Recall that Gurobi is the underlying LP solver. We choose the barrier method and disable crossover by setting Gurobi's parameters: `Method=2` and `Crossover=0`. In addition, we solve the test instances with two barrier convergence tolerances by changing Gurobi's parameter `BarTol`. We keep the default settings of Gurobi otherwise. Tables EC.13 and EC.14 in Appendix EC.3.1 report the times (in seconds) for Gurobi to terminate for the hotel instances with `BarTol=1e-6`. Table 2 shows the average running times of the three formulations and DEC in its last column.

We plot the average solution times and the number of non-optimal instances in Figure 4. On the left, the barrier convergence tolerance is 1e-6. On the right, the barrier convergence tolerance is 1e-7. We can see that (P) and ($\mathrm{D}_\mathbf{c}$) have a significant advantage over (R). First, (P) and ($\mathrm{D}_\mathbf{c}$) have better computational efficiency than (R). When `BarTol=1e-6`, the average computation time of (R) is 38.2 seconds, while that of (P) and ($\mathrm{D}_\mathbf{c}$) are 32.1 and 27.0 seconds, respectively. Thus, on average, (P) and ($\mathrm{D}_\mathbf{c}$) are 1.19 and 1.42 times faster than (R), respectively.

Second, (P) and ($\mathrm{D}_\mathbf{c}$) have better numerical stability. When `BarTol=1e-6`, both (P) and ($\mathrm{D}_\mathbf{c}$) can terminate with "Optimal" status for all 192 instances. However, (R) does not terminate with "Optimal" status for three out of the 192 instances and instead terminates with "Suboptimal" status. This is troublesome because Gurobi does not provide a dual solution needed to construct the policy when it terminates with "Suboptimal" status. Therefore, (P) and ($\mathrm{D}_\mathbf{c}$) have better numerical performance than (R), even though they are larger formulations than (R). With modern optimization tools, it is not rare for a larger formulation to outperform a smaller one; many examples are documented in Bertsimas and Dunn (2019). Among (P) and ($\mathrm{D}_\mathbf{c}$), ($\mathrm{D}_\mathbf{c}$) appears to have better computational performance because it has a shorter running time and terminates with "Optimal" status for all 192 instances.

We further test (P) and ($\mathrm{D}_\mathbf{c}$) with a tighter barrier convergence tolerance: `BarTol=1e-7`. The detailed results are included in Tables EC.15 and EC.16 in Appendix EC.3.2. Under this barrier convergence tolerance, the average times of (P) and ($\mathrm{D}_\mathbf{c}$) are 39.9 and 51.5 seconds, respectively, as shown in Figure 4. Even though (P) has shorter running times than ($\mathrm{D}_{\mathbf{c}}$) on average, (P) terminates with "Suboptimal" status for 25 instances, while ($\mathrm{D}_{\mathbf{c}}$) does so for only five instances. Thus, ($\mathrm{D}_{\mathbf{c}}$) is able to solve 20 more instances to optimality than (P) with a tighter convergence tolerance. This again demonstrates the effectiveness of the product-value inequalities on providing robust computational performance.

The detailed running times of DEC are in Table EC.17. Overall, the average running time of DEC is 13.5 seconds in our implementation. Although DEC has longer solution times than ($\mathrm{D}_{\mathbf{c}}$) for the hub-and-spoke instances, as shown in Table 2, DEC can be further sped up in several ways. Among them, parallelization might be the most significant one because DEC can be parallelized by resources. However, DEC obtains much looser upper bounds on the expected revenue. To evaluate the policy performance, one still needs to solve the resource-based or product-based SPL approximation. The advantage of the product-based SPL approximation is that it provides more granular information and allows for better interpretability.

## 5. Conclusion

In this paper, we propose a product-based ALP approach for the NRM problem. We start with a new policy referred to as the product-based bid-price policy, which takes the network effect into account and alleviates the non-optimality of resource-based bid-price policies (Talluri and van Ryzin 1998). To facilitate the product-based bid-price policy, we propose a product-based SPL approximation to capture more granular information than the resource-based SPL approximation. We show that the resulting ALP problem admits a compact equivalent reformulation, which we call the reduced program. We also derive valid inequalities, which can further improve the policy and computational performance.

We conduct numerical experiments on 192 problem instances considering different network structures, selling horizons, and load factors. The product-based bid-price policy significantly outperforms the resource-based one, with an average revenue gain of 0.72% and a maximum gain of 5.3%. The product-based SPL approximation can be solved 1.42 times faster than the resource-based SPL approximation. It also has better numerical stability and obtains optimal solutions for 20 more instances. Adding the valid inequalities is critical for improving the expected revenue and enhancing the computational performance. Our best implementation of the product-based bid-price policy shows stronger performance than the dynamic programming decomposition heuristic for the hub-and-spoke instances.

There are several avenues for future research. First, we assume independent demand throughout the paper. It is natural to incorporate customer choice models (Vulcano et al. 2010, Berbeglia et al. 2021). While the product-based SPL approximation can be applied, it is unclear whether a compact reformulation exists and whether the resulting ALP can be efficiently solved. Establishing valid inequalities and their impact is another potential challenge. We might even push this direction further to consider the setting in which the demand parameters are unknown (see Topaloglu and Powell 2006, Besbes and Zeevi 2012). Second, unlike the existing ALP approaches that are motivated by developing stronger functional approximations and tighter bounds, our research starts by constructing good policies, and then the approximation architecture is chosen to match the proposed policies. It would be interesting to explore this idea for other application settings. Lastly, we use valid inequalities to guide the selection of optimal solutions to improve both the policy and computational performance. This opens opportunities to further investigate creative applications of classic mathematical programming theory to LP-based ADPs.

## Acknowledgments

The authors thank the area editor Gustavo Vulcano, an anonymous associate editor, and two anonymous referees for their insightful comments and suggestions. The work of the second author was partially supported by the Canadian Natural Science and Engineering Research Council (NSERC RGPIN-2020-04229).

## References

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

Ben Amor, H., Desrosiers, J., and Valério de Carvalho, J. M. (2006). Dual-optimal inequalities for stabilized column generation. *Operations Research*, 54(3):454–463.

Berbeglia, G., Garassino, A., and Vulcano, G. (2021). A comparative empirical study of discrete choice models in retail operations. *Management Science*.

Bertsimas, D. and Dunn, J. W. (2019). *Machine learning under a modern optimization lens*. Dynamic Ideas LLC.

Besbes, O. and Zeevi, A. (2012). Blind network revenue management. *Operations Research*, 60(6):1537–1550.

Bezanson, J., Edelman, A., Karpinski, S., and Shah, V. B. (2017). Julia: A fresh approach to numerical computing. *SIAM Review*, 59(1):65–98.

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

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

de Farias, D. and Van Roy, B. (2004). On constraint sampling in the linear programming approach to approximate dynamic programming. *Mathematics of Operations Research*, 29(3):462–478.

de Farias, D. P. and Van Roy, B. (2003). The linear programming approach to approximate dynamic programming. *Operations Research*, 51(6):850–865.

Dunning, I., Huchette, J., and Lubin, M. (2017). JuMP: A modeling language for mathematical optimization. *SIAM Review*, 59(2):295–320.

Farias, V. F. and Van Roy, B. (2007). An approximate dynamic programming approach to network revenue mangement. Working paper, MIT Sloan School of Management.

Gallego, G., Topaloglu, H., et al. (2019). *Revenue management and pricing analytics*, volume 209. Springer.

Gallego, G. and van Ryzin, G. J. (1997). A multiproduct dynamic pricing problem and its applications to network yield management. *Operations Research*, 45(1):24–41.

Gschwind, T. and Irnich, S. (2016). Dual inequalities for stabilized column generation revisited. *INFORMS Journal on Computing*, 28(1):175–194.

Gurobi Optimization (2022). Gurobi Optimizer Reference Manual.

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

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

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

Meissner, J. and Strauss, A. K. (2012). Network revenue management with inventory-sensitive bid prices and customer choice. *European Journal of Operational Research*, 216(2):459–468.

Nadarajah, S., Margot, F., and Secomandi, N. (2015). Relaxations of approximate linear programs for the real option management of commodity storage. *Management Science*, 61(12):3054–3076.

Nadarajah, S. and Secomandi, N. (2020). Least squares Monte Carlo and approximate linear programming: Error bounds and energy real option application. Now Publishers.

Nemhauser, G. and Wolsey, L. (1988). *Integer and Combinatorial Optimization*. Wiley, New York.

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

Talluri, K. and van Ryzin, G. J. (2004). *The Theory and Practice of Revenue Management*. Springer, New York, NY.

Tong, C. and Topaloglu, H. (2014). 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.

Topaloglu, H. and Powell, W. B. (2006). Dynamic-programming approximations for stochastic time-staged integer multicommodity-flow problems. *INFORMS Journal on Computing*, 18(1):31–42.

Van Ryzin, G. (2005). Future of revenue management. *Journal of Revenue and Pricing Management*, 4(2):204–210.

Vossen, T. and Zhang, D. (2015a). A dynamic disaggregation approach to approximate linear programs for network revenue management. *Production and Operations Management*, 24(3):488–503.

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

Vulcano, G., van Ryzin, G., and Chaar, W. (2010). Choice-based revenue management: An empirical study of estimation and optimization. *Manufacturing & Service Operations Management*, 12(3):371–392.

Williamson, E. (1992). *Airline Network Seat Control*. PhD thesis, Massachusetts Institute of Technology.

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

---

**Rui Zhang** is Assistant Professor of Operations Management at Leeds School of Business, University of Colorado Boulder. His research interests are in quantitative methods, especially prescriptive analytics techniques. His work focuses on developing novel methods for revenue management problems, routing problems in logistics, and influence maximization problems on social networks.

**Saied Samiedaluie** is Assistant Professor in the Department of Accounting and Business Analytics at the Alberta School of Business, University of Alberta. His research interests focus on developing approximate dynamic programming methods for revenue management and studying data-driven analytical models to support decision-making in healthcare.

**Dan Zhang** is Professor of Operations Management at Leeds School of Business, University of Colorado Boulder, and Changjiang Chaired Professor awarded by the Chinese Ministry of Education. Dr. Zhang's primary research interest is revenue management and approximate dynamic programming.

# Electronic Companion

## EC.1. Technical Proofs

### EC.1.1. Proof of Theorem 1

Plugging (17) into the formulation (LP) and simplifying the constraints, we obtain

$$
\begin{aligned}
(\mathrm{D}^{\underline{W}})\ \min_{\theta,\underline{W}}\quad & \theta_1+\sum_j \mathbb{1}\left(\mathbf{x}\geq \mathbf{a}^j\right)\sum_{i\in\mathbf{a}^j}\sum_{k=1}^{x_i}\underline{W}_{1,i,j,k}\\
\text{s.t.}\quad & \theta_t+\sum_j \mathbb{1}\left(\mathbf{x}\geq \mathbf{a}^j\right)\sum_{i\in\mathbf{a}^j}\sum_{k=1}^{x_i}\underline{W}_{t,i,j,k}\\
&\geq \sum_j \lambda_{t,j}u_{t,j}\left(f_j+\theta_{t+1}+\sum_m \mathbb{1}\left(\mathbf{x}-\mathbf{a}^j\geq \mathbf{a}^m\right)\sum_{i\in\mathbf{a}^m}\sum_{k=1}^{x_i}\underline{W}_{t+1,i,j,k}\right)\\
&\quad+\left(1-\sum_j \lambda_{t,j}u_{t,j}\right)\left(\theta_{t+1}+\sum_j \mathbb{1}\left(\mathbf{x}\geq \mathbf{a}^j\right)\sum_{i\in\mathbf{a}^j}\sum_{k=1}^{x_i}\underline{W}_{t+1,i,j,k}\right),\quad \forall t,\mathbf{x}\in\mathcal{X}_t,\mathbf{u}_t\in\mathcal{U}(\mathbf{x}).
\end{aligned}
\tag{EC.1}
$$

Given an optimal solution $(\theta^*,\underline{W}^*)$ to $(\mathrm{D}^{\underline{W}})$, we can approximate the value functions as:

$$
v_t(\mathbf{x})\approx v_t^{\underline{W}}(\mathbf{x})=\theta_t^*+\sum_j \mathbb{1}\left(\mathbf{x}\geq \mathbf{a}^j\right)\sum_{i\in\mathbf{a}^j}\sum_{k=1}^{x_i}\underline{W}^*_{t,i,j,k},\quad \forall t,\mathbf{x}\in\mathcal{X}_t. \tag{EC.2}
$$

Recall that the resource-based SPL value function approximation is denoted by $v_t^R(\mathbf{x})$. In what follows, we show that, like $v_t^R(\mathbf{x})$, $v_t^{\underline{W}}(\mathbf{x})$ provides state-wise upper bounds of $v_t(\mathbf{x})$. Moreover, $v_t^{\underline{W}}(\mathbf{x})$ provides state-wise lower bounds of $v_t^R(\mathbf{x})$. Therefore, $v_t^{\underline{W}}(\mathbf{x})$ has state-wise smaller approximation errors than $v_t^R(\mathbf{x})$.

First, we show that $v_t^{\underline{W}}(\mathbf{x})$ is an upper bound of $v_t(\mathbf{x})$ for all $t$ and $\mathbf{x}\in\mathcal{X}_t$.

Proposition EC.1. *$v_t^{\underline{W}}(\mathbf{x})\geq v_t(\mathbf{x})$ for all $t$, $\mathbf{x}\in\mathcal{X}_t$.*

*Proof:* We prove the result by induction. Consider $(\mathrm{D}')$ and $t=T$, we note from (EC.1) that

$$
\theta_T^*+\sum_j \mathbb{1}\left(\mathbf{x}\geq \mathbf{a}^j\right)\sum_{i\in\mathbf{a}^j}\sum_{k=1}^{x_i}\underline{W}^*_{T,i,j,k}\geq \sum_j \lambda_{T,j}u_{T,j}f_j,\quad \mathbf{x}\in\mathcal{X}_T,\mathbf{u}_T\in\mathcal{U}(\mathbf{x}).
$$

It follows that

$$
\theta_T^*+\sum_j \mathbb{1}\left(\mathbf{x}\geq \mathbf{a}^j\right)\sum_{i\in\mathbf{a}^j}\sum_{k=1}^{x_i}\underline{W}^*_{T,i,j,k}\geq \max_{\mathbf{u}_T\in\mathcal{U}(\mathbf{x})}\sum_j \lambda_{T,j}u_{T,j}f_j=v_T(\mathbf{x}),\quad \mathbf{x}\in\mathcal{X}_T,.
$$

Thus, the result holds for $t=T$. Next, assume the result holds for $t+1$. Consider $(\mathrm{D}^{\underline{W}})$ and the inductive assumption for all $\mathbf{x}\in\mathcal{X}_{t+1}$, we have

$$
\theta_t^*+\sum_j \mathbb{1}\left(\mathbf{x}\geq \mathbf{a}^j\right)\sum_{i\in\mathbf{a}^j}\sum_{k=1}^{x_i}\underline{W}^*_{t,i,j,k}
$$

$$
\begin{aligned}
&\geq \sum_j \lambda_{t,j} u_{t,j} \left( f_j + \theta^*_{t+1} + \sum_m \mathbb{1}\left(\mathbf{x} - \mathbf{a}^j \geq \mathbf{a}^m\right) \sum_{i \in \mathbf{a}^m} \sum_{k=1}^{x_i} \underline{W}^*_{t+1,i,j,k} \right) \\
&\quad + \left(1 - \sum_j \lambda_{t,j} u_{t,j}\right) \left( \theta^*_{t+1} + \sum_j \mathbb{1}\left(\mathbf{x} \geq \mathbf{a}^j\right) \sum_{i \in \mathbf{a}^j} \sum_{k=1}^{x_i} \underline{W}^*_{t+1,i,j,k} \right) \\
&\geq \sum_j \lambda_{t,j} u_{t,j} \left( f_j + v_{t+1}(\mathbf{x} - \mathbf{a}^j) \right) + \left(1 - \sum_j \lambda_{t,j} u_{t,j}\right) v_{t+1}(\mathbf{x}), \quad \mathbf{x} \in \mathcal{X}_t, \mathbf{u}_t \in \mathcal{U}(\mathbf{x}).
\end{aligned}
$$

Again, it follows that

$$
\begin{aligned}
&\theta^*_t + \sum_j \mathbb{1}\left(\mathbf{x} \geq \mathbf{a}^j\right) \sum_{i \in \mathbf{a}^j} \sum_{k=1}^{x_i} \underline{W}^*_{t,i,j,k} \\
\geq &\max_{\mathbf{u}_t \in \mathcal{U}(\mathbf{x})} \sum_j \lambda_{t,j} u_{t,j} \left( f_j + v_{t+1}(\mathbf{x} - \mathbf{a}^j) \right) + \left(1 - \sum_j \lambda_{t,j} u_{t,j}\right) v_{t+1}(\mathbf{x}) \\
= &v_t(\mathbf{x}), \qquad \mathbf{x} \in \mathcal{X}_t.
\end{aligned}
$$

The last equality follows the optimality equation for $t$. $\square$

Now, we construct a state-wise upper bound of $v_t(\mathbf{x})$ and show that this upper bound is equal to $v_t^R(\mathbf{x})$ later. Given an optimal solution $(W^*, \beta^*, \gamma^*, \delta^*)$ to (D) and Theorem 2, $W^*$ is also optimal to (D′), with $\theta_t = 0$ for all $t$. Following the product-based SPL approximation, we can approximate the value function as:

$$
v_t(\mathbf{x}) \approx \sum_i \sum_{j \in \mathbf{a}_i} \sum_{k=1}^{x_i} W^*_{t,i,j,k}, \quad \forall t, \mathbf{x} \in \mathcal{X}_t.
$$

In this way, we also obtain an upper bound of $v_t(\mathbf{x})$ for all $t$ and $\mathbf{x} \in \mathcal{X}_T$.

Proposition EC.2. *$\sum_i \sum_{j \in \mathbf{a}_i} \sum_{k=1}^{x_i} W^*_{t,i,j,k} \geq v_t(\mathbf{x})$ for all $t$, $\mathbf{x} \in \mathcal{X}_t$.*

*Proof:* We prove the result by induction. Consider (D′) and $t = T$, we note from (19) that

$$
\sum_j \sum_{i \in \mathbf{a}^j} \sum_{k=1}^{x_i} W^*_{T,i,j,k} \geq \sum_j \lambda_{T,j} u_{T,j} f_j, \quad \mathbf{x} \in \mathcal{X}_T, \mathbf{u}_T \in \mathcal{U}(\mathbf{x}).
$$

It follows that

$$
\sum_j \sum_{i \in \mathbf{a}^j} \sum_{k=1}^{x_i} W^*_{T,i,j,k} \geq \max_{\mathbf{u}_T \in \mathcal{U}(\mathbf{x})} \sum_j \lambda_{T,j} u_{T,j} f_j = v_T(\mathbf{x}), \quad \mathbf{x} \in \mathcal{X}_T.
$$

Thus, the result holds for $t = T$. Next, assume the result holds for $t+1$. Consider (D′) and the inductive assumption for all $\mathbf{x} \in \mathcal{X}_{t+1}$, we have

$$
\sum_j \sum_{i \in \mathbf{a}^j} \sum_{k=1}^{x_i} W^*_{t,i,j,k}
$$

$$\geq \sum_j \lambda_{t,j} u_{t,j} \left( f_j + \sum_j \sum_{i\in\mathbf{a}^j} \sum_{k=1}^{x_i - a_{i,j}} W^*_{t+1,i,j,k} \right) + \left( 1 - \sum_j \lambda_{t,j} u_{t,j} \right) \sum_j \sum_{i\in\mathbf{a}^j} \sum_{k=1}^{x_i} W^*_{t+1,i,j,k}$$
$$\geq \sum_j \lambda_{t,j} u_{t,j} \left( f_j + v_{t+1}(\mathbf{x} - \mathbf{a}^j) \right) + \left( 1 - \sum_j \lambda_{t,j} u_{t,j} \right) v_{t+1}(\mathbf{x}), \quad \mathbf{x} \in \mathcal{X}_t, \mathbf{u}_t \in \mathcal{U}(\mathbf{x}).$$

Again, it follows that

$$\sum_j \sum_{i\in\mathbf{a}^j} \sum_{k=1}^{x_i} W^*_{t,i,j,k} \geq \max_{\mathbf{u}_t \in \mathcal{U}(\mathbf{x})} \sum_j \lambda_{t,j} u_{t,j} \left( f_j + v_{t+1}(\mathbf{x} - \mathbf{a}^j) \right) + \left( 1 - \sum_j \lambda_{t,j} u_{t,j} \right) v_{t+1}(\mathbf{x})$$
$$= v_t(\mathbf{x}), \qquad \mathbf{x} \in \mathcal{X}_t.$$

The last equality follows the optimality equation for $t$. $\square$

Next, we show that $\sum_i \sum_{j\in\mathbf{a}_i} \sum_{k=1}^{x_i} W^*_{t,i,j,k} \geq v^{\underline{W}}_t(\mathbf{x})$ for all $t$ and $\mathbf{x} \in \mathcal{X}_t$. This result helps us compare the approximation error between these two approximations, $v^{\underline{W}}_t(\mathbf{x})$ and $v^R_t(\mathbf{x})$.

Proposition EC.3. *$\sum_i \sum_{j\in\mathbf{a}_i} \sum_{k=1}^{x_i} W^*_{t,i,j,k} \geq v^{\underline{W}}_t(\mathbf{x})$ for all $t$, $\mathbf{x} \in \mathcal{X}_t$.*

*Proof:* We prove the result by induction. Without loss of generality, we consider an optimal solution to ($\mathrm{D}^{\underline{W}}$) with $\theta^*_t = 0$ for all $t$. Let $t = T$, given (19) and (EC.1), we have $\sum_{i\in\mathbf{a}^j} W^*_{T,i,j,1} = \sum_{i\in\mathbf{a}^j} \underline{W}^*_{T,i,j,1} = \lambda_{t,j} f_j$ and $W^*_{T,i,j,k} = \underline{W}^*_{T,i,j,k} = 0$ for each $j$ in $\mathcal{J}$, $i$ in $\mathbf{a}^j$ and $1 < k \leq c_i$. Therefore, we have

$$\sum_j \sum_{i\in\mathbf{a}^j} \sum_{k=1}^{x_i} W^*_{T,i,j,k} \geq \sum_j \mathbb{1}\left( \mathbf{x} \geq \mathbf{a}^j \right) \sum_{i\in\mathbf{a}^j} \sum_{k=1}^{x_i} \underline{W}^*_{T,i,j,k} = v^{\underline{W}}_T(\mathbf{x}), \quad \mathbf{x} \in \mathcal{X}_T.$$

Thus, the result holds for $t = T$. Next, assume the result holds for $t+1$. Consider ($\mathrm{D}'$), ($\mathrm{D}^{\underline{W}}$) and the inductive assumption for all $\mathbf{x} \in \mathcal{X}_{t+1}$, we have

$$\sum_j \sum_{i\in\mathbf{a}^j} \sum_{k=1}^{x_i} W^*_{t,i,j,k}$$
$$\geq \sum_j \lambda_{t,j} u_{t,j} \left( f_j + \sum_m \sum_{i\in\mathbf{a}^m} \sum_{k=1}^{x_i - a_{i,j}} W^*_{t+1,i,m,k} \right) + \left( 1 - \sum_j \lambda_{t,j} u_{t,j} \right) \sum_j \sum_{i\in\mathbf{a}^j} \sum_{k=1}^{x_i} W^*_{t+1,i,j,k} \tag{EC.3}$$

and

$$\sum_j \mathbb{1}\left( \mathbf{x} \geq \mathbf{a}^j \right) \sum_{i\in\mathbf{a}^j} \sum_{k=1}^{x_i} \underline{W}^*_{t,i,j,k} \geq \sum_j \lambda_{t,j} u_{t,j} \left( f_j + v^{\underline{W}}_{t+1}(\mathbf{x} - \mathbf{a}^j) \right) + \left( 1 - \sum_j \lambda_{t,j} u_{t,j} \right) v^{\underline{W}}_{t+1}(\mathbf{x}) \tag{EC.4}$$

The right-hand side of (EC.3) is larger than that of (EC.4). Thus, it follows that

$$\sum_j \sum_{i\in\mathbf{a}^j} \sum_{k=1}^{x_i} W^*_{t,i,j,k} \geq \sum_j \mathbb{1}\left( \mathbf{x} \geq \mathbf{a}^j \right) \sum_{i\in\mathbf{a}^j} \sum_{k=1}^{x_i} \underline{W}^*_{t,i,j,k} = v^{\underline{W}}_t(\mathbf{x}), \quad \mathbf{x} \in \mathcal{X}_t.$$

The first inequality holds because both ($\mathrm{D}'$) and ($\mathrm{D}^{\underline{W}}$) are minimization problems. $\square$

Now, we show that $\sum_j \sum_{i\in\mathbf{a}^j} \sum_{k=1}^{x_i} W^*_{t,i,j,k} = v^R_t(\mathbf{x})$ for all $t$, $\mathbf{x} \in \mathcal{X}_t$. Notice that $\sum_j \sum_{i\in\mathbf{a}^j} \sum_{k=1}^{x_i} W^*_{t,i,j,k}$ is based on $W^*_{t,i,j,k}$, while $v^R_t(\mathbf{x})$ needs $V^*_{t,i,k}$. We can relate $\sum_j \sum_{i\in\mathbf{a}^j} \sum_{k=1}^{x_i} W^*_{t,i,j,k}$ and $v^R_t(\mathbf{x})$ as follows. On the one hand, given an optimal solution $W^*_{t,i,j,k}$ to (D), we can recover $v^R_t(\mathbf{x})$ by following (13), i.e, $V^*_{t,i,k} = \sum_{j\in\mathbf{a}_i} W^*_{t,i,j,k}$ for all $t,i,k$. On the other hand, the dual of (R) is

$$
\begin{aligned}
\text{(RD)} \quad \min_{V,\beta,\gamma,\delta} \quad & \sum_i \sum_{k=1}^{c_i} V_{1,i,k} \\
\text{s.t.} \quad & V_{t,i,k} - V_{t+1,i,k} = \sum_{j\in\mathbf{a}_i} \delta_{t,i,j,k}, && \forall t,i,k, && \text{(EC.5)} \\
& \sum_{i\in\mathbf{a}^j} \beta_{t,i,j} = \lambda_{t,j} f_j, && \forall t,j, && \text{(EC.6)} \\
& \delta_{t,i,j,k} - \gamma_{t,i,j,k} + \lambda_{t,j} V_{t+1,i,k} \\
& \qquad = \begin{cases} \beta_{t,i,j}, & \text{if } k=1, \\ \lambda_{t,j} V_{t+1,i,k-1} - \gamma_{t,i,j,k-1}, & \text{if } k \ge 2, \end{cases} && \forall t,i,j,k : a_{i,j}=1, && \text{(EC.7)} \\
& \gamma_{t,i,j,k}, \delta_{t,i,j,k} \ge 0, && \forall t,i,j,k : a_{i,j}=1. && \text{(EC.8)}
\end{aligned}
$$

Given an optimal solution $V^*_{t,i,k}$ to (RD), setting $\tilde{W}_{t,i,j,k} = V^*_{t,i,k}/|\mathbf{a}^j|$ for all $t,i,j,k$ gives an optimal solution to (D) because all constraints in (D) are satisfied, and the objective value of (D) with $\tilde{W}_{t,i,j,k}$ is the same as the optimal objective value of (RD) with $V^*_{t,i,k}$. Thus, $\sum_j \sum_{i\in\mathbf{a}^j} \sum_{k=1}^{x_i} W^*_{t,i,j,k} = v^R_t(\mathbf{x})$ for all $t$, $\mathbf{x} \in \mathcal{X}_t$.

We have shown that both $v^W_t(\mathbf{x})$ and $\sum_j \sum_{i\in\mathbf{a}^j} \sum_{k=1}^{x_i} W^*_{t,i,j,k}$ are the upper bound of $v_t(\mathbf{x})$ for all $t$ and $\mathbf{x} \in \mathcal{X}_T$ in Propositions EC.1 and EC.2. Furthermore, we know that $v^W_t(\mathbf{x}) \le \sum_j \sum_{i\in\mathbf{a}^j} \sum_{k=1}^{x_i} W^*_{t,i,j,k}$ for all $t$, $\mathbf{x} \in \mathcal{X}_t$ by Proposition EC.3. Finally, we know how to relate $\sum_j \sum_{i\in\mathbf{a}^j} \sum_{k=1}^{x_i} W^*_{t,i,j,k}$ and $v^R_t(\mathbf{x})$ such that they are equal to each other. Putting it together, we have the desired result in Theorem 1. $\square$

### EC.1.2. Proof of Theorem 2

Because (P) is derived from (P′) via variable aggregation, every feasible solution of (P′) corresponds to a feasible solution to (P) with the same objective value.

To establish equivalence, we must show that each feasible solution to (P), denoted by $(\mathbf{q}^*, \mathbf{p}^*, \mathbf{z}^*)$, corresponds to a feasible solution to (P′), denoted by $\mathbf{h}^*$. Denote period $t$ values of $(\mathbf{q}^*, \mathbf{p}^*, \mathbf{z}^*)$ as $(\mathbf{q}^*_t, \mathbf{p}^*_t, \mathbf{z}^*_t)$. Define $\mathcal{P}_t = \{q_{t,j} = z_{t,i,j,1}\ \forall i,j : a_{i,j} = 1, z_{t,i,j,k+1} \le z_{t,i,j,k}\ \forall i,j,k : a_{i,j} = 1, z_{t,i,j,k} \le p_{t,i,k}\ \forall i,j,k : a_{i,j} = 1, 0 \le \mathbf{q}_t, \mathbf{p}_t, \mathbf{z}_t \le 1\}$ for a given $t$. $\mathcal{P}_t$ is an integral polytope because the matrix is a totally unimodular one. Each row of this matrix has at most one +1, and at most one -1. As a result, the constraint matrix is totally unimodular from Propositions 2.1 and 2.6 in Nemhauser and Wolsey (1988, p.540 and p. 542). Therefore, all extreme points of $\mathcal{P}_t$ are integral. Thus, $(\mathbf{q}^*_t, \mathbf{p}^*_t, \mathbf{z}^*_t)$ is a convex combination of the extreme points of $\mathcal{P}_t$. Denote an extreme point of $\mathcal{P}_t$ as $(\mathbf{q}^e_t, \mathbf{p}^e_t, \mathbf{z}^e_t)$

indexed by $e$, its weight in a convex combination as $h_t^e$ and the set of extreme points of $\mathcal{P}_t$ as $\mathcal{E}_t$. Then, $(\mathbf{q}_t^*, \mathbf{p}_t^*, \mathbf{z}_t^*)$ can be represented as

$$
\begin{aligned}
q_{t,j}^* &= \sum_{e \in E_t} q_{t,j}^e h_t^e, && \forall t, j, \\
p_{t,i,k}^* &= \sum_{e \in E_t} p_{t,i,k}^e h_t^e, && \forall t, i, k, \\
z_{t,j,i,k}^* &= \sum_{e \in E_t} p_{t,i,k}^e q_{t,j}^e h_t^e, && \forall t, j, i, k : a_{ij} = 1, \\
\sum_{e \in E_t} h_t^e &= 1, \\
h_t^e &\geq 0, && \forall e \in E_t.
\end{aligned}
$$

Based on the definition of $\mathcal{P}_t$, $(\mathbf{q}_t^e, \mathbf{p}_t^e, \mathbf{z}_t^e)$ corresponds to a point in the state-action space, and $\mathcal{E}_t$ is equivalent to the state-action space. Furthermore, we can obtain the value of $u_{t,j}$ by the value of $q_{t,j}^e$ for each $j$ in $\mathcal{J}$ and the value of $x_{t,i}$ by the value of $p_{t,i,k}^e$ for each $i$ in $\mathcal{I}$ from an extreme point $e$. Then, we can set $h_{t,\mathbf{x},\mathbf{u}}^* = h_t^e$ for all corresponding state-action pairs and extreme points $e$ in $\mathcal{E}_t$ and repeat the argument for all $t$ in $\mathcal{T}$. Thus, each feasible solution to (P) has a corresponding solution to (P′). This completes the proof. $\square$

### EC.1.3. Proof of Proposition 1

The inequality $z_R \geq v_1(\mathbf{c})$ is well-known in the literature. We only need to show the equality. It can be verified that constraints (23), (24), and (25) of (P) are identical to constraints (7), (8), and (9) of (R). Furthermore, constraint (22) adds $|\mathbf{a}_i|$ copies of constraint (6) for each resource $i$. As a result, $z_P = z_R$. $\square$

### EC.1.4. Proof of Lemma 1

We first show that an optimal solution that does not satisfy (30) can be modified to satisfy the condition. Then, we further modify the solution to satisfy (31). Finally, we invoke a known result in Vossen and Zhang (2015b) to establish (32).

*Step 1:* Suppose we are given an optimal solution $(\tilde{W}, \tilde{\beta}, \tilde{\gamma}, \tilde{\delta})$ that does not satisfy (30). We can construct a new optimal solution $(W^*, \tilde{\beta}, \tilde{\gamma}, \tilde{\delta})$ that satisfies (30) as follows.

Let $t^{'}$ be the largest time index for which (30) is violated for some $i^{'}$, $j^{'}$, and $k^{'}$. We define two product sets:

$$
\begin{aligned}
\mathbf{a}_{t',i',k'}^{+} &= \left\{ j \in \mathbf{a}_i : \tilde{W}_{t',i',j,k'} - \tilde{W}_{t'+1,i',j,k'} > 0 \right\}, \\
\mathbf{a}_{t',i',k'}^{-} &= \left\{ j \in \mathbf{a}_i : \tilde{W}_{t',i',j,k'} - \tilde{W}_{t'+1,i',j,k'} < 0 \right\}.
\end{aligned}
$$

We construct $W^*$ such that $\sum_{j \in \mathbf{a}_i} W_{t',i',j,k'}^* = \sum_{j \in \mathbf{a}_i} \tilde{W}_{t',i',j,k'}$ as follows:

- For each $j$ in $\mathbf{a}^-_{t',i',k'}$, set $W^*_{t',i',j,k'} = \tilde{W}_{t'+1,i',j,k'}$. Let $\epsilon = -\sum_{j\in \mathbf{a}^-_{t',i',k'}} \left(\tilde{W}_{t',i',j,k'} - \tilde{W}_{t'+1,i',j,k'}\right)$.
- For each $j$ in $\mathbf{a}^+_{t',i',k'}$, set $W^*_{t',i',j,k'} = \tilde{W}_{t',i',j,k'} + \min\{\epsilon, \tilde{W}_{t',i',j,k'} - \tilde{W}_{t'+1,i',j,k'}\}$ and update $\epsilon = \epsilon - (W^*_{t',i',j,k'} - \tilde{W}_{t',i',j,k'})$.

From (26) and (29), we must have

$$\sum_{j\in\mathbf{a}_i} (W_{t,i,j,k} - W_{t+1,i,j,k}) \geq 0, \quad \forall t,i,k.$$

Therefore,

$$\sum_{j\in\mathbf{a}^+_{t',i',k'}} \left(\tilde{W}_{t',i',j,k'} - \tilde{W}_{t'+1,i',j,k'}\right) > \sum_{j\in\mathbf{a}^-_{t',i',k'}} \left(\tilde{W}_{t',i',j,k'} - \tilde{W}_{t'+1,i',j,k'}\right).$$

Hence, we are able to allocate the deficit over the products in $\mathbf{a}^-_{t',i',k'}$ to those in $\mathbf{a}^+_{t',i',k'}$. By construction, $\sum_{j\in\mathbf{a}_i} W^*_{t',i',j,k'} = \sum_{j\in\mathbf{a}_i} \tilde{W}_{t',i',j,k'}$, which ensures that (28) is maintained. Successively constructing the values backwards in time yields a feasible solution $(W^*, \tilde{\beta}, \tilde{\gamma}, \tilde{\delta})$ that has the same objective value as that of $(\tilde{W}, \tilde{\beta}, \tilde{\gamma}, \tilde{\delta})$.

*Step 2:* Given that (30) holds, we further modify the solution $(W^*, \tilde{\beta}, \tilde{\gamma}, \tilde{\delta})$ to obtain another solution $(W^*, \beta^*, \gamma^*, \delta^*)$ with the same objective value as follows.

First, we set $\beta^* = \tilde{\beta}$ and

$$\delta^*_{t,i,j,k} = W^*_{t,i,j,k} - W^*_{t+1,i,j,k}, \quad \forall t,i,j,k : a_{i,j} = 1.$$

Then, for each $t,i,j,k$ with $a_{i,j}=1$, take

$$\begin{aligned}
\gamma^*_{t,i,j,k} &= \begin{cases} \delta^*_{t,i,j,k} + \lambda_{t,j} \sum_{m\in\mathbf{a}_i} W^*_{t+1,i,m,k} - \beta^*_{t,i,j}, & \text{if } k=1, \\ \delta^*_{t,i,j,k} + \lambda_{t,j} \sum_{m\in\mathbf{a}_i} W^*_{t+1,i,m,k} \\ \quad -\lambda_{t,j} \sum_{m\in\mathbf{a}_i} W^*_{t+1,i,m,k-1} + \gamma^*_{t,i,j,k-1}, & \text{if } k\geq 2, \end{cases} \\
&= \sum_{k'=1}^{k} \delta^*_{t,i,j,k'} + \lambda_{t,j} \sum_{m\in\mathbf{a}_i} W^*_{t+1,i,m,k} - \beta^*_{t,i,j}, \\
&= \sum_{k'=1}^{k} (W^*_{t,i,j,k'} - W^*_{t+1,i,j,k'}) + \lambda_{t,j} \sum_{m\in\mathbf{a}_i} W^*_{t+1,i,m,k} - \beta^*_{t,i,j}. \qquad \text{(EC.9)}
\end{aligned}$$

The second equality is obtained by recursively substituting out $\gamma^*_{t,i,j,k-1}$ for $k\geq 2$. The third equality follows from substituting out $\delta^*_{t,i,j,k'}$. Next, we modify the value of $\beta^*$ in the following way to ensure the non-negativity of $\gamma^*$. For each $t$ and $j$,

- Let $\rho_{t,j} = \lambda_{t,j} f_j$.
- For each $i$ in $\mathbf{a}^j$, let

$$\sigma_{t,i,j} = \min_{k\in\{1,\cdots,c_i\}} \left\{ \sum_{k'=1}^{k} (W^*_{t,i,j,k'} - W^*_{t+1,i,j,k'}) + \lambda_{t,j} \sum_{m\in\mathbf{a}_i} W^*_{t+1,i,m,k} \right\},$$

set $\beta^*_{t,i,j} = \min\{\rho_{t,j}, \sigma_{t,i,j}\}$ and update $\rho_{t,j} = \rho_{t,j} - \beta^*_{t,i,j}$.

We still must show that $\beta^*$ satisfies constraint (27). To achieve this, we show that $\sum_{i\in\mathbf{a}^j}\sigma_{t,i,j}$ is greater than or equal to $\lambda_{t,j}f_j$ for the given $t$ and $j$. Furthermore, we know that $\sigma_{t,i,j}\geq 0$ because $W^*_{t,i,j,k}\geq W^*_{t+1,i,j,k}\geq 0$. Therefore, we can allocate the value of $\lambda_{t,j}f_j$ to $\beta^*_{t,i,j}$ over $i\in\mathbf{a}^j$ by the constructed solution. Note that $\beta^*_{t,i,j}\geq 0$ by construction.

Given $t$ and $j$, let

$$x^*_{t,i,j}=\underset{k\in\{1,\cdots,c_i\}}{\arg\min}\left\{\sum_{k'=1}^{k}(W^*_{t,i,j,k'}-W^*_{t+1,i,j,k'})+\lambda_{t,j}\sum_{m\in\mathbf{a}_i}W^*_{t+1,i,m,k}\right\}$$

for each $i\in\mathbf{a}^j$. Based on Theorem 2 and the strong duality of linear programming, (D) is equivalent to (D$'$). This implies that $W^*$ is an optimal solution to (D$'$) with $\theta_t=0$ for all $t$. Thus, for a state-action pair $(\mathbf{x},\mathbf{u}_t)$ with $x_i=x^*_{t,i,j}$, $u_{t,j}=1$ for each $i$ in $\mathbf{a}^j$, and $u_{t,j}=0$ for other entries. Hence, we have

$$\sum_{i\in\mathbf{a}^j}\left\{\sum_{k=1}^{x^*_{t,i,j}}(W^*_{t,i,j,k}-W^*_{t+1,i,j,k})+\lambda_{t,j}\sum_{m\in\mathbf{a}_i}W^*_{t+1,i,m,x^*_i}\right\}\geq\lambda_{t,j}f_j,$$

where the inequality follows from constraint (19) for the given state-action pair $(\mathbf{x},\mathbf{u}_t)$. Therefore, we can ensure that constraint (27) is satisfied by the constructed solution.

The solution $(W^*,\beta^*,\gamma^*,\delta^*)$ has the same objective value as that of $(\tilde{W},\tilde{\beta},\tilde{\gamma},\tilde{\delta})$. Thus, it is optimal and satisfies (30) and (31).

*Step 3:* Lemma 5 in Vossen and Zhang (2015b) shows that the resource-based SPL approximation has an optimal solution that satisfies $V_{t,i,k}\geq V_{t,i,k+1}$ for all $t,i,k$. Given (13), $V_{t,i,k}=\sum_{j\in\mathbf{a}_i}W_{t,i,j,k}$ for all $t,i,k$, we have $\sum_{j\in\mathbf{a}_i}W^*_{t,i,j,k}\geq\sum_{j\in\mathbf{a}_i}W^*_{t,i,j,k+1}$ for all $t,i,k$. $\square$

### EC.1.5. Proof of Proposition 2

Without loss of optimality, we focus on solutions that satisfy the conditions in Lemma 1. Since $W$ is monotone in $t$ according to (30), we have

$$W_{t,i,j,k}-W_{t+1,i,j,k}\geq 0,\quad\forall t,i,j,k.$$

Therefore, if (33) holds for $\mathbf{x}=\mathbf{c}$, then it holds for all $\mathbf{x}$. In the remainder of the proof, we focus on $\mathbf{x}=\mathbf{c}$.

From Lemma 1, we can replace (26) by

$$W_{t,i,j,k}-W_{t+1,i,j,k}=\delta_{t,i,j,k},\quad\forall t,i,j,k.$$

Substituting out $\delta$, (28) can be rewritten as

$$\begin{aligned}&W_{t,i,j,k}-W_{t+1,i,j,k}-\gamma_{t,i,j,k}+\lambda_{t,j}\sum_{m\in\mathbf{a}_i}W_{t+1,i,m,k}\\&\quad=\begin{cases}\beta_{t,i,j}, & \text{if } k=1,\\ \lambda_{t,j}\sum\limits_{m\in\mathbf{a}_i}W_{t+1,i,m,k-1}-\gamma_{t,i,j,k-1}, & \text{if } k\geq 2,\end{cases}\qquad\forall t,i,j,k:a_{i,j}=1.\end{aligned}$$

Suppose (33) does not hold for a given optimal solution $(\tilde{W}, \tilde{\beta}, \tilde{\gamma}, \tilde{\delta})$. Let $t'$ be the largest time index for which there is a product $j'$ violating (33); i.e.,

$$\sum_{i \in \mathbf{a}^{j'}} \sum_{k=1}^{c_i} \tilde{W}_{t',i,j',k} - \sum_{i \in \mathbf{a}^{j'}} \sum_{k=1}^{c_i} \tilde{W}_{t'+1,i,j',k} > \lambda_{t',j'} f_{j'}.$$

We construct an optimal solution $(W^*, \beta^*, \gamma^*, \delta^*)$ that satisfies (33) as follows:

- Initialize $(W^*, \beta^*, \gamma^*, \delta^*) = (\tilde{W}, \tilde{\beta}, \tilde{\gamma}, \tilde{\delta})$.
- For each $i$ in $\mathbf{a}^{j'}$, substituting out $\gamma^*_{t',i,j',k-1}$ recursively for $k=1$ to $c_i$ and using $\sum_{k'=1}^{k-1} W^*_{t',i,j',k'} - \sum_{k'=1}^{k-1} \tilde{W}_{t'+1,i,j',k'} \geq 0$ yields (EC.9). Based on it, we set

$$W^*_{t',i,j',k} = \max\left\{0, \tilde{\beta}_{t',i,j'} - \lambda_{t',j'} \sum_{m \in \mathbf{a}_i} \tilde{W}_{t'+1,i,m,k}\right\} - \sum_{k'=1}^{k-1} (W^*_{t',i,j',k'} - \tilde{W}_{t'+1,i,j',k'}) + \tilde{W}_{t'+1,i,j',k}, \tag{EC.10}$$

$$\delta^*_{t',i,j',k} = \begin{cases} 0, & \text{if } \tilde{\beta}_{t',i,j'} - \lambda_{t',j'} \sum_{m \in \mathbf{a}_i} \tilde{W}_{t'+1,i,m,k} \leq 0, \\ \tilde{\beta}_{t',i,j'} - \lambda_{t',j'} \sum_{m \in \mathbf{a}_i} \tilde{W}_{t'+1,i,m,k} - \sum_{k'=1}^{k-1} (W^*_{t',i,j',k'} - \tilde{W}_{t'+1,i,j',k'}), & \text{otherwise}, \end{cases}$$
$$= \begin{cases} 0, & \text{if } \tilde{\beta}_{t',i,j'} - \lambda_{t',j'} \sum_{m \in \mathbf{a}_i} \tilde{W}_{t'+1,i,m,k} \leq 0, \\ \tilde{\beta}_{t',i,j'} - \lambda_{t',j'} \sum_{m \in \mathbf{a}_i} \tilde{W}_{t'+1,i,m,k} - \max\left\{0, \tilde{\beta}_{t',i,j'} - \lambda_{t',j'} \sum_{m \in \mathbf{a}_i} \tilde{W}_{t'+1,i,m,k-1}\right\}, & \text{otherwise}, \end{cases}$$

$$\gamma^*_{t',i,j',k} = (W^*_{t',i,j',k} - \tilde{W}_{t'+1,i,j',k}) - (\tilde{\beta}_{t',i,j'} - \lambda_{t',j'} \sum_{m \in \mathbf{a}_i} \tilde{W}_{t'+1,i,m,k}) + \sum_{k'=1}^{k-1} (W^*_{t',i,j',k'} - \tilde{W}_{t'+1,i,j',k'})$$
$$= \begin{cases} -(\tilde{\beta}_{t',i,j'} - \lambda_{t',j'} \sum_{m \in \mathbf{a}_i} \tilde{W}_{t'+1,i,m,k}), & \text{if } \tilde{\beta}_{t',i,j'} - \lambda_{t',j'} \sum_{m \in \mathbf{a}_i} \tilde{W}_{t'+1,i,m,k} \leq 0, \\ 0, & \text{otherwise}. \end{cases}$$

To verify the nonnegativity of $\delta^*$ and $\gamma^*$, we rely on the property stated in (32), i.e., $\sum_{j \in \mathbf{a}_i} \tilde{W}_{t'+1,i,j',k} \geq \sum_{j \in \mathbf{a}_i} \tilde{W}_{t'+1,i,j',k+1}$ for all $i, k$. Given a $k$, we have $\tilde{\beta}_{t',i,j'} - \lambda_{t',j'} \sum_{m \in \mathbf{a}_i} \tilde{W}_{t'+1,i,m,k} \geq \tilde{\beta}_{t',i,j'} - \lambda_{t',j'} \sum_{m \in \mathbf{a}_i} \tilde{W}_{t'+1,i,m,k'}$ for all $k' \leq k$. Thus, if $\tilde{\beta}_{t',i,j'} - \lambda_{t',j'} \sum_{m \in \mathbf{a}_i} \tilde{W}_{t'+1,i,m,k} \geq 0$, $\delta^*_{t',i,j',k}$ is nonnegative due to $\sum_{k'=1}^{k-1} (\tilde{W}_{t',i,j',k'} - \tilde{W}_{t'+1,i,j',k'}) = \max\left\{0, \tilde{\beta}_{t',i,j'} - \lambda_{t',j'} \sum_{m \in \mathbf{a}_i} \tilde{W}_{t'+1,i,m,k-1}\right\}$ and $\max\left\{0, \tilde{\beta}_{t',i,j'} - \lambda_{t',j'} \sum_{m \in \mathbf{a}_i} \tilde{W}_{t'+1,i,m,k}\right\} \geq \max\left\{0, \tilde{\beta}_{t',i,j'} - \lambda_{t',j'} \sum_{m \in \mathbf{a}_i} \tilde{W}_{t'+1,i,m,k-1}\right\}$. If $\tilde{\beta}_{t',i,j'} - \lambda_{t',j'} \sum_{m \in \mathbf{a}_i} \tilde{W}_{t'+1,i,m,k} \leq 0$, $\gamma^*_{t',i,j',k}$ is nonnegative because $\tilde{\beta}_{t',i,j'} - \lambda_{t',j'} \sum_{m \in \mathbf{a}_i} \tilde{W}_{t'+1,i,m,k'} \leq 0$ and $W^*_{t',i,j',k'} = \tilde{W}_{t'+1,i,j',k'}$ for all $k' \leq k$.

By construction, (30) and (31) are satisfied between $W^*_{t',i,j',k}$ and $W^*_{t'+1,i,j',k}$ for all $i$ and $k$. Furthermore, using the same way to set $W^*$ at $t'-1$ for all $i$ and $k$, we have the following due to $W^*_{t'-1,i,j',k} - W^*_{t',i,j',k} \geq 0$ for all $i$ and $k$:

$$\sum_{k'=1}^{k+1} (W^*_{t'-1,i,j',k} - W^*_{t',i,j',k}) - \sum_{k'=1}^{k} (W^*_{t'-1,i,j',k} - W^*_{t',i,j',k}) \geq 0.$$

Following (EC.10), we have

$$\max\left\{0,\tilde{\beta}_{t'-1,i,j'}-\lambda_{t'-1,j'}\sum_{m\in\mathbf{a}_i}W^*_{t',i,m,k+1}\right\}-\max\left\{0,\tilde{\beta}_{t'-1,i,j'}-\lambda_{t'-1,j'}\sum_{m\in\mathbf{a}_i}W^*_{t',i,m,k}\right\}\geq 0.$$

This leads to

$$\sum_{m\in\mathbf{a}_i}W^*_{t',i,m,k}-\sum_{m\in\mathbf{a}_i}W^*_{t',i,m,k+1}\geq 0$$

Notice that $W^*_{t',i,j',k}$ are not affected by $W^*_{t'-1,i,j',k}$. Thus, (32) is satisfied by $W^*$ at $t^{'}$.

Furthermore, $\sum_{k'=1}^{k}W^*_{t',i,j',k'}\leq\sum_{k'=1}^{k}\tilde{W}_{t',i,j',k'}$ for all $i$ in $\mathbf{a}^{j^{'}}$ and $k$. Recall that $\tilde{W}$ satisfies Lemma 1, which means for all $i$ in $\mathbf{a}^{j^{'}}$ and $k$, following (EC.9) and the nonnegativity of $\gamma$ yields

$$\sum_{k'=1}^{k}(\tilde{W}_{t',i,j',k'}-\tilde{W}_{t'+1,i,j',k'})+\lambda_{t',j'}\sum_{m\in\mathbf{a}_i}\tilde{W}_{t'+1,i,m,k}-\tilde{\beta}_{t',i,j'}\geq 0.$$

Combining with $\sum_{k'=1}^{k}(\tilde{W}_{t',i,j',k'}-\tilde{W}_{t'+1,i,j',k'})\geq 0$ gives

$$\sum_{k'=1}^{k}(\tilde{W}_{t',i,j',k'}-\tilde{W}_{t'+1,i,j',k'})\geq\max\left\{0,\tilde{\beta}_{t',i,j'}-\lambda_{t',j'}\sum_{m\in\mathbf{a}_i}\tilde{W}_{t'+1,i,m,k}\right\}$$

Moving the term $(-\sum_{k'=1}^{k}\tilde{W}_{t'+1,i,j',k'})$ to the right-hand side gives

$$\begin{aligned}\sum_{k'=1}^{k}\tilde{W}_{t',i,j',k}&\geq\max\left\{0,\tilde{\beta}_{t',i,j'}-\lambda_{t',j'}\sum_{m\in\mathbf{a}_i}\tilde{W}_{t'+1,i,m,k}\right\}+\sum_{k'=1}^{k}\tilde{W}_{t'+1,i,j',k'}\\&=\sum_{k'=1}^{k}W^*_{t',i,j',k}\end{aligned}$$

The last equality is due to the construction of $W^*$. Thus, at $t^{'}$, $\sum_{k'=1}^{k}W^*_{t',i,j,k'}\leq\sum_{k'=1}^{k}\tilde{W}_{t',i,j,k'}$ for all $j$, $i$ in $\mathbf{a}^j$ and $k$. Then, using the same way to set $W^*$ for product $j^{'}$ at $t^{'}-1$ for all $i\in\mathbf{a}^j$ and $k$ gives

$$\begin{aligned}&\sum_{k'=1}^{k}\tilde{W}_{t'-1,i,j',k}\\&\geq\max\left\{0,\tilde{\beta}_{t'-1,i,j'}-\lambda_{t'-1,j'}\sum_{m\in\mathbf{a}_i}\tilde{W}_{t',i,m,k}\right\}+\sum_{k'=1}^{k}\tilde{W}_{t',i,j',k'}\\&=\max\left\{\sum_{k'=1}^{k}\tilde{W}_{t',i,j',k'},\tilde{\beta}_{t'-1,i,j'}-\lambda_{t'-1,j'}\sum_{m\in\mathbf{a}_i}\tilde{W}_{t',i,m,k}+\sum_{k'=1}^{k}\tilde{W}_{t',i,j',k'}\right\}\\&=\max\left\{\sum_{k'=1}^{k}\tilde{W}_{t',i,j',k'},\right.\end{aligned}$$

$$\tilde{\beta}_{t'-1,i,j'} - \lambda_{t'-1,j'} \sum_{m\in\mathbf{a}_i\setminus j'} \tilde{W}_{t',i,m,k} + (1-\lambda_{t'-1,j'}) \sum_{k'=1}^{k} \tilde{W}_{t',i,j',k'} + \lambda_{t'-1,j'} \sum_{k'=1}^{k-1} \tilde{W}_{t',i,j',k'} \Bigg\}$$

$$\geq \max\left\{ \sum_{k'=1}^{k} W^*_{t',i,j',k'}, \tilde{\beta}_{t'-1,i,j'} - \lambda_{t'-1,j'} \sum_{m\in\mathbf{a}_i} W^*_{t',i,m,k} + \sum_{k'=1}^{k} W^*_{t',i,j',k'} \right\}$$

$$= \max\left\{ 0, \tilde{\beta}_{t'-1,i,j'} - \lambda_{t'-1,j'} \sum_{m\in\mathbf{a}_i} W^*_{t',i,m,k} \right\} + \sum_{k'=1}^{k} W^*_{t',i,j',k'}$$

$$= \sum_{k'=1}^{k} W^*_{t'-1,i,j',k}$$

The first inequality follows because $\tilde{W}$ satisfies Lemma 1. The second inequality is due to $0 \leq \lambda_{t'-1,j'} \leq 1$ and $\sum_{k'=1}^{k} W^*_{t',i,j',k'} \leq \sum_{k'=1}^{k} \tilde{W}_{t',i,j',k'}$ for all $i$ in $\mathbf{a}^{j'}$ and $k$. The last equality is due to the construction of $W^*$. Thus, $\sum_{k'=1}^{k} W^*_{t,i,j',k'} \leq \sum_{k'=1}^{k} \tilde{W}_{t,i,j',k'}$ for all $t \leq t'$, $i$ in $\mathbf{a}^{j'}$ and $k$. It leads to $\sum_{k'=1}^{k} W^*_{t,i,j,k} \leq \sum_{k'=1}^{k} \tilde{W}_{t,i,j,k}$ for all $t$, $i$, $j$, and $k$.

Now, let $\mathcal{A}^{j'}_* = \left\{ i \in \mathbf{a}^{j'} : \sum_{k=1}^{c_i} (W^*_{t',i,j',k} - W^*_{t'+1,i,j',k}) = \beta^*_{t',i,j'} - \lambda_{t',j'} \sum_{m\in\mathbf{a}_i} W^*_{t'+1,i,m,k} \right\}$, we have

$$\begin{aligned}
&\sum_{i\in\mathbf{a}^{j'}} \sum_{k=1}^{c_i} W^*_{t',i,j',k} - \sum_{i\in\mathbf{a}^{j'}} \sum_{k=1}^{c_i} W^*_{t'+1,i,j',k} \\
=& \sum_{i\in\mathbf{a}^{j'}} \sum_{k=1}^{c_i} (W^*_{t',i,j',k} - W^*_{t'+1,i,j',k}) \\
=& \sum_{i\in\mathbf{a}^{j'}} \max\left\{ 0, \beta^*_{t',i,j'} - \lambda_{t',j'} \sum_{m\in\mathbf{a}_i} W^*_{t'+1,i,m,c_i} \right\} \\
=& \sum_{i\in\mathcal{A}^{j'}_*} (\beta^*_{t',i,j'} - \lambda_{t',j'} \sum_{m\in\mathbf{a}_i} W^*_{t'+1,i,m,c_i}) \\
\leq& \sum_{i\in\mathbf{a}^{j'}} \beta^*_{t',i,j'} - \sum_{i\in\mathcal{A}^{j'}_*} \lambda_{t',j'} \sum_{m\in\mathbf{a}_i} W^*_{t'+1,i,m,c_i} \\
\leq& \lambda_{t',j'} f_{j'}
\end{aligned}$$

The second to the last inequality follows because $\beta^*_{t',i,j'} \geq 0$ as shown in Step 2 of the proof of Lemma 1. The last inequality is due to the nonnegativity of $W$ and (27), i.e., $\sum_{i\in\mathbf{a}^j} \beta^*_{t',i,j'} = \lambda_{t',j'} f_{j'}$.

Thus, successively constructing the values backwards in time yields a feasible solution $(W^*, \beta^*, \gamma^*, \delta^*)$ that satisfies (33). Moreover, $\sum_{k'=1}^{k} W^*_{t,i,j,k} \leq \sum_{k'=1}^{k} \tilde{W}_{t,i,j,k}$ for all $t$, $i$, $j$, and $k$ by construction. Therefore, $\sum_j \sum_{i\in\mathbf{a}^j} \sum_{k'=1}^{c_i} W^*_{1,i,j,k} \leq \sum_j \sum_{i\in\mathbf{a}^j} \sum_{k'=1}^{c_i} \tilde{W}_{1,i,j,k}$, the objective value of $(W^*, \beta^*, \gamma^*, \delta^*)$ is less than or equal to that of $(\tilde{W}, \tilde{\beta}, \tilde{\gamma}, \tilde{\delta})$. Therefore, either we obtain an optimal solution that satisfies (33) or we find the contradiction that $(\tilde{W}, \tilde{\beta}, \tilde{\gamma}, \tilde{\delta})$ is not an optimal solution. This completes the proof. $\square$

## EC.2. Detailed Results on Policy Performance

There are two kinds of tables in this section. For the ones reporting policy performance, the first column shows the parameters $(T, I, \kappa, \alpha)$ for each hotel instance or $(T, N, \kappa, \alpha)$ for each hub-and-spoke instance. The second to fifth columns show the means and standard errors (SEs) obtained by RB, PB, $PB_{\mathbf{c}}$ and DEC over 100,000 sample paths. The last column gives the upper bound on the optimal expected revenue, which is the optimal objective value of (P).

For the ones reporting comparison between policies, the first column shows the parameters $(I, \kappa, \alpha)$ for each hotel instance or $(N, \kappa, \alpha)$ for each hub-and-spoke instance. Then, the results are grouped by the number of periods $(T)$. We show the percentage gains of PB and $PB_{\mathbf{c}}$ over RB. We also report the percentage gaps of RB and $PB_{\mathbf{c}}$ between the average revenues and the upper bound as $PB_{\mathbf{c}}$ has the best performance. Finally, we show the average revenues obtained by DEC and the percent difference comparing $PB_{\mathbf{c}}$ to DEC, which is calculated as ($PB_{\mathbf{c}}$ - DEC)/DEC.

### EC.2.1. Policy Performance on Hotel Instances

| Parameters | RB | | PB | | $\mathrm{PB_c}$ | | DEC | | Upp. |
|---|---|---|---|---|---|---|---|---|---|
| $(T, I, \kappa, \alpha)$ | Mean | SE | Mean | SE | Mean | SE | Mean | SE | Bnd. |
| (50,2,4,1.6) | 5053 | 2.1 | 5062 | 2.1 | 5120 | 2.0 | 5164 | 2.0 | 5174 |
| (50,2,4,2.2) | 6458 | 2.1 | 6478 | 2.1 | 6555 | 2.1 | 6581 | 2.1 | 6634 |
| (50,2,4,3.0) | 4658 | 2.1 | 4666 | 2.1 | 4728 | 2.0 | 4746 | 2.0 | 4794 |
| (50,3,4,1.6) | 6273 | 2.3 | 6276 | 2.3 | 6289 | 2.3 | 6309 | 2.4 | 6419 |
| (50,3,4,2.2) | 3846 | 1.6 | 3851 | 1.6 | 3941 | 1.5 | 4041 | 1.5 | 4092 |
| (50,3,4,3.0) | 2327 | 1.1 | 2329 | 1.1 | 2335 | 1.1 | 2353 | 1.1 | 2398 |
| (50,4,4,1.6) | 3192 | 1.5 | 3197 | 1.5 | 3214 | 1.5 | 3233 | 1.6 | 3297 |
| (50,4,4,2.2) | 1584 | 0.8 | 1588 | 0.7 | 1593 | 0.7 | 1640 | 0.8 | 1672 |
| (50,4,4,3.0) | 3364 | 2.0 | 3395 | 2.0 | 3401 | 2.0 | 3488 | 2.1 | 3561 |
| (50,5,4,1.6) | 8260 | 3.2 | 8211 | 3.2 | 8286 | 3.2 | 8370 | 3.5 | 8626 |
| (50,5,4,2.2) | 6979 | 3.2 | 7004 | 3.2 | 7021 | 3.2 | 7126 | 3.5 | 7381 |
| (50,5,4,3.0) | 6009 | 3.0 | 6040 | 3.0 | 6028 | 2.9 | 6107 | 3.3 | 6361 |
| (50,2,8,1.6) | 6410 | 3.3 | 6421 | 3.3 | 6439 | 3.3 | 6453 | 3.3 | 6456 |
| (50,2,8,2.2) | 10235 | 4.9 | 10245 | 4.9 | 10472 | 5.0 | 10609 | 4.9 | 10651 |
| (50,2,8,3.0) | 4251 | 2.3 | 4250 | 2.3 | 4326 | 2.2 | 4349 | 2.2 | 4355 |
| (50,3,8,1.6) | 6257 | 3.9 | 6270 | 3.9 | 6279 | 3.9 | 6314 | 4.1 | 6361 |
| (50,3,8,2.2) | 6139 | 4.0 | 6128 | 4.1 | 6168 | 4.0 | 6230 | 4.0 | 6253 |
| (50,3,8,3.0) | 5990 | 4.1 | 5991 | 4.1 | 6012 | 4.0 | 6061 | 4.1 | 6090 |
| (50,4,8,1.6) | 12567 | 7.0 | 12615 | 7.0 | 12607 | 7.0 | 12746 | 7.4 | 12928 |
| (50,4,8,2.2) | 8545 | 5.1 | 8566 | 5.0 | 8590 | 5.0 | 8685 | 5.1 | 8771 |
| (50,4,8,3.0) | 4261 | 2.7 | 4261 | 2.7 | 4271 | 2.7 | 4277 | 2.9 | 4344 |
| (50,5,8,1.6) | 5508 | 3.8 | 5541 | 3.8 | 5534 | 3.8 | 5603 | 4.1 | 5692 |
| (50,5,8,2.2) | 12674 | 8.1 | 12704 | 8.0 | 12703 | 8.0 | 12827 | 8.7 | 13029 |
| (50,5,8,3.0) | 13529 | 7.3 | 13526 | 7.3 | 13564 | 7.3 | 13566 | 7.7 | 13948 |

Table EC.1: Policy Performance (Mean and Standard Error (SE)) for Hotel Instances with 50 Periods Based on 100,000 Sample Paths

| Parameters | RB | | PB | | $\text{PB}_\mathbf{c}$ | | DEC | | Upp. |
|---|---|---|---|---|---|---|---|---|---|
| $(T, I, \kappa, \alpha)$ | Mean | SE | Mean | SE | Mean | SE | Mean | SE | Bnd. |
| (100,2,4,1.6) | 10704 | 2.1 | 10713 | 2.1 | 10942 | 2.0 | 10993 | 2.0 | 11066 |
| (100,2,4,2.2) | 4896 | 1.5 | 4905 | 1.5 | 5061 | 1.4 | 5166 | 1.3 | 5171 |
| (100,2,4,3.0) | 1277 | 0.4 | 1277 | 0.4 | 1277 | 0.4 | 1283 | 0.4 | 1290 |
| (100,3,4,1.6) | 10856 | 3.2 | 10866 | 3.1 | 10898 | 3.1 | 10953 | 3.2 | 11068 |
| (100,3,4,2.2) | 4685 | 1.8 | 4696 | 1.8 | 4716 | 1.8 | 4744 | 1.8 | 4782 |
| (100,3,4,3.0) | 2771 | 1.0 | 2779 | 1.0 | 2776 | 1.0 | 2806 | 1.0 | 2835 |
| (100,4,4,1.6) | 9304 | 2.9 | 9318 | 2.8 | 9340 | 2.8 | 9421 | 2.9 | 9518 |
| (100,4,4,2.2) | 9171 | 3.0 | 9219 | 2.9 | 9243 | 2.9 | 9385 | 3.1 | 9542 |
| (100,4,4,3.0) | 6166 | 2.7 | 6200 | 2.7 | 6200 | 2.7 | 6300 | 2.9 | 6394 |
| (100,5,4,1.6) | 13773 | 3.6 | 13800 | 3.6 | 13835 | 3.5 | 14021 | 3.6 | 14247 |
| (100,5,4,2.2) | 12665 | 4.5 | 12689 | 4.4 | 12721 | 4.4 | 12985 | 4.6 | 13244 |
| (100,5,4,3.0) | 8865 | 3.7 | 8876 | 3.7 | 8933 | 3.7 | 9101 | 3.9 | 9283 |
| (100,2,8,1.6) | 17316 | 6.4 | 17315 | 6.4 | 17435 | 6.3 | 17496 | 6.3 | 17516 |
| (100,2,8,2.2) | 17266 | 4.5 | 17266 | 4.5 | 17566 | 4.7 | 17696 | 4.7 | 17773 |
| (100,2,8,3.0) | 11011 | 3.1 | 11014 | 3.1 | 11083 | 3.2 | 11111 | 3.2 | 11134 |
| (100,3,8,1.6) | 29776 | 7.9 | 29761 | 7.9 | 29941 | 8.0 | 30151 | 8.2 | 30324 |
| (100,3,8,2.2) | 19427 | 7.6 | 19478 | 7.5 | 19532 | 7.5 | 19633 | 7.8 | 19763 |
| (100,3,8,3.0) | 11462 | 6.4 | 11491 | 6.3 | 11501 | 6.3 | 11549 | 6.5 | 11629 |
| (100,4,8,1.6) | 14214 | 6.0 | 14266 | 5.9 | 14327 | 5.9 | 14449 | 6.2 | 14559 |
| (100,4,8,2.2) | 19789 | 6.8 | 19800 | 6.7 | 19883 | 6.8 | 20063 | 7.0 | 20247 |
| (100,4,8,3.0) | 15995 | 6.4 | 16022 | 6.4 | 16021 | 6.4 | 16119 | 6.8 | 16299 |
| (100,5,8,1.6) | 21463 | 8.2 | 21492 | 8.2 | 21512 | 8.2 | 21671 | 8.9 | 21942 |
| (100,5,8,2.2) | 22072 | 8.4 | 22086 | 8.4 | 22117 | 8.3 | 22324 | 8.7 | 22539 |
| (100,5,8,3.0) | 17447 | 8.3 | 17492 | 8.3 | 17500 | 8.3 | 17620 | 8.6 | 17814 |

Table EC.2: Policy Performance (Mean and Standard Error (SE)) for Hotel Instances with 100 Periods Based on 100,000 Sample Paths

| Parameters | RB | | PB | | $\text{PB}_{\mathbf{c}}$ | | DEC | | Upp. |
|---|---|---|---|---|---|---|---|---|---|
| $(T, I, \kappa, \alpha)$ | Mean | SE | Mean | SE | Mean | SE | Mean | SE | Bnd. |
| (200,2,4,1.6) | 32184 | 4.9 | 32220 | 4.9 | 32550 | 4.7 | 32622 | 4.7 | 32730 |
| (200,2,4,2.2) | 20689 | 4.7 | 20745 | 4.6 | 21585 | 3.9 | 21653 | 3.9 | 21711 |
| (200,2,4,3.0) | 20933 | 3.3 | 20948 | 3.3 | 21159 | 3.5 | 21201 | 3.6 | 21290 |
| (200,3,4,1.6) | 39848 | 5.7 | 39876 | 5.7 | 39977 | 5.7 | 40642 | 5.8 | 40898 |
| (200,3,4,2.2) | 23584 | 5.2 | 23644 | 5.1 | 24091 | 4.8 | 24212 | 4.9 | 24329 |
| (200,3,4,3.0) | 11965 | 3.5 | 12006 | 3.5 | 12139 | 3.3 | 12229 | 3.4 | 12322 |
| (200,4,4,1.6) | 24527 | 4.5 | 24526 | 4.5 | 24681 | 4.4 | 24908 | 4.5 | 25085 |
| (200,4,4,2.2) | 34357 | 6.2 | 34354 | 6.2 | 34575 | 6.2 | 34931 | 6.5 | 35279 |
| (200,4,4,3.0) | 10504 | 3.7 | 10533 | 3.7 | 10524 | 3.7 | 10666 | 3.8 | 10802 |
| (200,5,4,1.6) | 27829 | 6.0 | 27829 | 5.9 | 27971 | 5.7 | 28095 | 5.9 | 28365 |
| (200,5,4,2.2) | 29550 | 7.9 | 29632 | 7.7 | 30615 | 7.0 | 31087 | 6.9 | 31411 |
| (200,5,4,3.0) | 23065 | 7.3 | 23125 | 7.2 | 23215 | 7.1 | 23571 | 7.3 | 23896 |
| (200,2,8,1.6) | 46332 | 8.1 | 46342 | 8.1 | 46385 | 8.2 | 46382 | 8.2 | 46495 |
| (200,2,8,2.2) | 36265 | 8.8 | 36310 | 8.7 | 37091 | 8.2 | 37188 | 8.1 | 37226 |
| (200,2,8,3.0) | 23219 | 5.2 | 23222 | 5.2 | 23246 | 5.2 | 23282 | 5.1 | 23332 |
| (200,3,8,1.6) | 31804 | 9.5 | 31907 | 9.4 | 32031 | 9.3 | 33187 | 8.2 | 33252 |
| (200,3,8,2.2) | 54543 | 14.9 | 54695 | 14.8 | 56523 | 13.5 | 56698 | 13.6 | 56837 |
| (200,3,8,3.0) | 22540 | 5.4 | 22538 | 5.4 | 22889 | 5.3 | 22977 | 5.5 | 23064 |
| (200,4,8,1.6) | 57057 | 13.1 | 57084 | 13.1 | 57255 | 13.0 | 57557 | 13.4 | 57820 |
| (200,4,8,2.2) | 41742 | 11.0 | 41782 | 11.0 | 42780 | 10.8 | 42921 | 11.0 | 43105 |
| (200,4,8,3.0) | 34856 | 8.7 | 34874 | 8.7 | 34937 | 8.8 | 35194 | 9.3 | 35442 |
| (200,5,8,1.6) | 32751 | 8.2 | 32782 | 8.2 | 32993 | 8.1 | 33174 | 8.5 | 33408 |
| (200,5,8,2.2) | 64819 | 16.1 | 64885 | 16.0 | 66023 | 15.6 | 66684 | 15.9 | 67013 |
| (200,5,8,3.0) | 42624 | 11.7 | 42654 | 11.7 | 42852 | 11.5 | 43224 | 12.0 | 43493 |

Table EC.3: Policy Performance (Mean and Standard Error (SE)) for Hotel Instances with 200 Periods Based on 100,000 Sample Paths

| Parameters | RB | | PB | | $\text{PB}_\mathbf{c}$ | | DEC | | Upp. |
|---|---|---|---|---|---|---|---|---|---|
| $(T, I, \kappa, \alpha)$ | Mean | SE | Mean | SE | Mean | SE | Mean | SE | Bnd. |
| (400,2,4,1.6) | 79995 | 8.7 | 80067 | 8.7 | 81459 | 7.4 | 82390 | 8.0 | 82588 |
| (400,2,4,2.2) | 30195 | 6.0 | 30249 | 6.0 | 31637 | 4.7 | 31726 | 4.5 | 31733 |
| (400,2,4,3.0) | 48166 | 6.6 | 48166 | 6.6 | 48467 | 6.4 | 48593 | 6.4 | 48705 |
| (400,3,4,1.6) | 39751 | 5.8 | 39756 | 5.8 | 39815 | 5.7 | 39977 | 5.6 | 40112 |
| (400,3,4,2.2) | 31171 | 5.5 | 31196 | 5.5 | 32237 | 4.6 | 32378 | 4.5 | 32472 |
| (400,3,4,3.0) | 60923 | 8.8 | 60956 | 8.8 | 61257 | 9.1 | 61586 | 9.2 | 61914 |
| (400,4,4,1.6) | 50554 | 7.4 | 50559 | 7.3 | 50822 | 7.1 | 51084 | 7.2 | 51381 |
| (400,4,4,2.2) | 61680 | 9.6 | 61740 | 9.6 | 63447 | 8.5 | 63719 | 8.5 | 64015 |
| (400,4,4,3.0) | 33688 | 6.3 | 33723 | 6.3 | 33839 | 6.1 | 34128 | 6.2 | 34393 |
| (400,5,4,1.6) | 61812 | 8.7 | 61760 | 8.6 | 62438 | 8.8 | 62883 | 9.0 | 63274 |
| (400,5,4,2.2) | 107527 | 15.3 | 107608 | 15.3 | 110220 | 14.2 | 111278 | 14.5 | 111984 |
| (400,5,4,3.0) | 41044 | 7.4 | 41078 | 7.3 | 41184 | 7.3 | 41587 | 7.7 | 41976 |
| (400,2,8,1.6) | 26287 | 3.6 | 26309 | 3.6 | 26601 | 3.9 | 27154 | 3.7 | 27205 |
| (400,2,8,2.2) | 27093 | 5.6 | 27127 | 5.6 | 27776 | 5.1 | 27812 | 5.1 | 27835 |
| (400,2,8,3.0) | 91855 | 14.1 | 91939 | 14.0 | 93581 | 14.3 | 93756 | 14.6 | 93881 |
| (400,3,8,1.6) | 126464 | 17.5 | 126462 | 17.4 | 126934 | 17.6 | 128212 | 18.8 | 128581 |
| (400,3,8,2.2) | 69251 | 11.9 | 69268 | 11.9 | 71121 | 11.3 | 71847 | 10.9 | 72019 |
| (400,3,8,3.0) | 52369 | 10.0 | 52346 | 10.0 | 52395 | 10.0 | 52472 | 10.0 | 52586 |
| (400,4,8,1.6) | 103514 | 15.8 | 103492 | 15.8 | 103816 | 15.7 | 104079 | 16.3 | 104472 |
| (400,4,8,2.2) | 157837 | 32.6 | 157858 | 32.6 | 166155 | 28.0 | 166582 | 28.7 | 167046 |
| (400,4,8,3.0) | 71153 | 14.5 | 71159 | 14.5 | 71413 | 14.2 | 71686 | 14.7 | 71906 |
| (400,5,8,1.6) | 86355 | 17.1 | 86435 | 17.0 | 87797 | 16.6 | 88127 | 16.9 | 88389 |
| (400,5,8,2.2) | 89108 | 16.4 | 89165 | 16.4 | 92457 | 15.9 | 92795 | 16.4 | 93098 |
| (400,5,8,3.0) | 56041 | 13.4 | 56050 | 13.3 | 56201 | 13.3 | 56536 | 13.7 | 56736 |

Table EC.4: Policy Performance (Mean and Standard Error (SE)) for Hotel Instances with 400 Periods Based on 100,000 Sample Paths

| | $T = 50$ | | | | | $T = 100$ | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Parameters | % Gain to RB | | % Gap to UB | | % $PB_c$ | % Gain to RB | | % Gap to UB | | % $PB_c$ |
| $(I, \kappa, \alpha)$ | PB | $PB_c$ | RB | $PB_c$ | To DEC | PB | $PB_c$ | RB | $PB_c$ | To DEC |
| (2,4,1.6) | 0.191 | 1.336 | 2.352 | 1.048 | -0.849 | 0.086 | 2.222 | 3.270 | 1.121 | -0.468 |
| (2,4,2.2) | 0.308 | 1.499 | 2.663 | 1.204 | -0.399 | 0.179 | 3.362 | 5.306 | 2.122 | -2.030 |
| (2,4,3.0) | 0.166 | 1.496 | 2.830 | 1.377 | -0.376 | 0.065 | 0.065 | 1.003 | 0.938 | -0.398 |
| (3,4,1.6) | 0.043 | 0.250 | 2.270 | 2.026 | -0.321 | 0.089 | 0.385 | 1.917 | 1.539 | -0.500 |
| (3,4,2.2) | 0.151 | 2.468 | 6.030 | 3.711 | -2.490 | 0.244 | 0.672 | 2.029 | 1.371 | -0.578 |
| (3,4,3.0) | 0.105 | 0.367 | 2.968 | 2.612 | -0.756 | 0.288 | 0.191 | 2.277 | 2.090 | -1.080 |
| (4,4,1.6) | 0.156 | 0.668 | 3.162 | 2.515 | -0.604 | 0.153 | 0.389 | 2.249 | 1.869 | -0.861 |
| (4,4,2.2) | 0.282 | 0.595 | 5.286 | 4.722 | -2.891 | 0.524 | 0.790 | 3.886 | 3.127 | -1.505 |
| (4,4,3.0) | 0.922 | 1.090 | 5.522 | 4.492 | -2.509 | 0.552 | 0.550 | 3.567 | 3.037 | -1.586 |
| (5,4,1.6) | -0.596 | 0.313 | 4.244 | 3.944 | -1.003 | 0.193 | 0.452 | 3.329 | 2.891 | -1.325 |
| (5,4,2.2) | 0.359 | 0.602 | 5.449 | 4.879 | -1.473 | 0.192 | 0.441 | 4.369 | 3.947 | -2.034 |
| (5,4,3.0) | 0.501 | 0.307 | 5.531 | 5.241 | -1.292 | 0.114 | 0.757 | 4.494 | 3.771 | -1.849 |
| (2,8,1.6) | 0.180 | 0.455 | 0.714 | 0.262 | -0.216 | -0.005 | 0.692 | 1.143 | 0.459 | -0.349 |
| (2,8,2.2) | 0.095 | 2.310 | 3.900 | 1.680 | -1.295 | 0.000 | 1.739 | 2.852 | 1.163 | -0.734 |
| (2,8,3.0) | -0.003 | 1.773 | 2.399 | 0.669 | -0.529 | 0.031 | 0.651 | 1.103 | 0.459 | -0.252 |
| (3,8,1.6) | 0.202 | 0.350 | 1.632 | 1.288 | -0.554 | -0.052 | 0.553 | 1.805 | 1.262 | -0.696 |
| (3,8,2.2) | -0.166 | 0.475 | 1.825 | 1.359 | -1.005 | 0.260 | 0.537 | 1.699 | 1.171 | -0.515 |
| (3,8,3.0) | 0.017 | 0.373 | 1.645 | 1.278 | -0.803 | 0.249 | 0.341 | 1.434 | 1.098 | -0.419 |
| (4,8,1.6) | 0.389 | 0.320 | 2.798 | 2.487 | -1.092 | 0.369 | 0.796 | 2.374 | 1.597 | -0.846 |
| (4,8,2.2) | 0.248 | 0.521 | 2.572 | 2.064 | -1.095 | 0.052 | 0.476 | 2.262 | 1.797 | -0.897 |
| (4,8,3.0) | 0.014 | 0.243 | 1.924 | 1.686 | -0.147 | 0.167 | 0.164 | 1.862 | 1.701 | -0.603 |
| (5,8,1.6) | 0.609 | 0.481 | 3.246 | 2.780 | -1.233 | 0.134 | 0.225 | 2.180 | 1.960 | -0.734 |
| (5,8,2.2) | 0.236 | 0.231 | 2.725 | 2.501 | -0.967 | 0.065 | 0.207 | 2.073 | 1.870 | -0.927 |
| (5,8,3.0) | -0.026 | 0.259 | 2.999 | 2.748 | -0.009 | 0.256 | 0.301 | 2.061 | 1.766 | -0.686 |

Table EC.5: Policy Comparison for Hotel Instances with 50 and 100 Periods Based on 100,000 Sample Paths

| | $T = 200$ | | | | | $T = 400$ | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Parameters | % Gain to RB | | % Gap to UB | | % $PB_{\mathbf{c}}$ | % Gain to RB | | % Gap to UB | | % $PB_{\mathbf{c}}$ |
| $(T, I, \kappa, \alpha)$ | PB | $PB_{\mathbf{c}}$ | RB | $PB_{\mathbf{c}}$ | To DEC | PB | $PB_{\mathbf{c}}$ | RB | $PB_{\mathbf{c}}$ | To DEC |
| (2,4,1.6) | 0.110 | 1.136 | 1.668 | 0.551 | -0.221 | 0.090 | 1.831 | 3.140 | 1.366 | -1.130 |
| (2,4,2.2) | 0.268 | 4.327 | 4.705 | 0.582 | -0.314 | 0.177 | 4.776 | 4.846 | 0.302 | -0.278 |
| (2,4,3.0) | 0.069 | 1.081 | 1.674 | 0.611 | -0.195 | -0.001 | 0.625 | 1.107 | 0.489 | -0.259 |
| (3,4,1.6) | 0.071 | 0.325 | 2.568 | 2.251 | -1.637 | 0.012 | 0.162 | 0.901 | 0.741 | -0.405 |
| (3,4,2.2) | 0.258 | 2.154 | 3.064 | 0.976 | -0.496 | 0.080 | 3.420 | 4.005 | 0.722 | -0.434 |
| (3,4,3.0) | 0.344 | 1.456 | 2.899 | 1.486 | -0.733 | 0.054 | 0.548 | 1.600 | 1.061 | -0.534 |
| (4,4,1.6) | -0.006 | 0.627 | 2.223 | 1.610 | -0.912 | 0.008 | 0.530 | 1.609 | 1.088 | -0.513 |
| (4,4,2.2) | -0.008 | 0.635 | 2.613 | 1.994 | -1.019 | 0.097 | 2.865 | 3.647 | 0.887 | -0.426 |
| (4,4,3.0) | 0.281 | 0.195 | 2.758 | 2.568 | -1.328 | 0.104 | 0.451 | 2.052 | 1.610 | -0.845 |
| (5,4,1.6) | -0.001 | 0.511 | 1.891 | 1.389 | -0.441 | -0.085 | 1.013 | 2.310 | 1.320 | -0.707 |
| (5,4,2.2) | 0.278 | 3.604 | 5.925 | 2.535 | -1.519 | 0.075 | 2.505 | 3.980 | 1.575 | -0.951 |
| (5,4,3.0) | 0.260 | 0.649 | 3.477 | 2.850 | -1.510 | 0.083 | 0.339 | 2.219 | 1.887 | -0.970 |
| (2,8,1.6) | 0.023 | 0.116 | 0.351 | 0.236 | 0.007 | 0.084 | 1.195 | 3.373 | 2.218 | -2.035 |
| (2,8,2.2) | 0.126 | 2.280 | 2.582 | 0.361 | -0.261 | 0.127 | 2.520 | 2.665 | 0.212 | -0.130 |
| (2,8,3.0) | 0.011 | 0.117 | 0.482 | 0.365 | -0.152 | 0.092 | 1.879 | 2.159 | 0.320 | -0.186 |
| (3,8,1.6) | 0.326 | 0.714 | 4.356 | 3.673 | -3.483 | -0.001 | 0.372 | 1.646 | 1.280 | -0.997 |
| (3,8,2.2) | 0.279 | 3.629 | 4.036 | 0.553 | -0.308 | 0.026 | 2.701 | 3.844 | 1.246 | -1.010 |
| (3,8,3.0) | -0.007 | 1.548 | 2.275 | 0.762 | -0.386 | -0.042 | 0.051 | 0.414 | 0.363 | -0.147 |
| (4,8,1.6) | 0.047 | 0.347 | 1.320 | 0.978 | -0.524 | -0.021 | 0.291 | 0.917 | 0.628 | -0.253 |
| (4,8,2.2) | 0.097 | 2.487 | 3.161 | 0.753 | -0.327 | 0.013 | 5.270 | 5.513 | 0.533 | -0.256 |
| (4,8,3.0) | 0.052 | 0.234 | 1.655 | 1.425 | -0.730 | 0.008 | 0.365 | 1.047 | 0.686 | -0.381 |
| (5,8,1.6) | 0.093 | 0.737 | 1.966 | 1.243 | -0.545 | 0.092 | 1.669 | 2.300 | 0.669 | -0.374 |
| (5,8,2.2) | 0.102 | 1.857 | 3.274 | 1.478 | -0.993 | 0.065 | 3.759 | 4.286 | 0.689 | -0.365 |
| (5,8,3.0) | 0.070 | 0.534 | 1.998 | 1.475 | -0.860 | 0.018 | 0.286 | 1.226 | 0.944 | -0.593 |

Table EC.6: Policy Comparison for Hotel Instances with 200 and 400 Periods Based on 100,000 Sample Paths

### EC.2.2. Policy Performance on Hub-and-Spoke Instances

| Parameters | RB | | PB | | $\text{PB}_\mathbf{c}$ | | DEC | | Upp. |
|---|---|---|---|---|---|---|---|---|---|
| $(T, N, \kappa, \alpha)$ | Mean | SE | Mean | SE | Mean | SE | Mean | SE | Bnd. |
| (50,2,4,1.6) | 1230 | 1.2 | 1244 | 1.2 | 1243 | 1.2 | 1249 | 1.3 | 1259 |
| (50,2,4,2.2) | 595 | 0.4 | 599 | 0.4 | 600 | 0.4 | 601 | 0.4 | 615 |
| (50,2,4,3.0) | 2772 | 1.5 | 2777 | 1.5 | 2783 | 1.5 | 2752 | 1.6 | 2833 |
| (50,3,4,1.6) | 2919 | 1.2 | 2918 | 1.2 | 2923 | 1.2 | 2897 | 1.3 | 3015 |
| (50,3,4,2.2) | 3472 | 1.7 | 3454 | 1.7 | 3485 | 1.7 | 3441 | 1.7 | 3575 |
| (50,3,4,3.0) | 2537 | 1.4 | 2537 | 1.4 | 2560 | 1.4 | 2535 | 1.5 | 2628 |
| (50,4,4,1.6) | 2799 | 1.3 | 2789 | 1.3 | 2806 | 1.3 | 2778 | 1.3 | 2920 |
| (50,4,4,2.2) | 2641 | 1.4 | 2638 | 1.4 | 2639 | 1.4 | 2633 | 1.5 | 2767 |
| (50,4,4,3.0) | 2196 | 1.4 | 2190 | 1.4 | 2201 | 1.4 | 2109 | 1.3 | 2297 |
| (50,5,4,1.6) | 4823 | 2.0 | 4807 | 1.9 | 4826 | 1.9 | 4760 | 2.0 | 5097 |
| (50,5,4,2.2) | 2796 | 1.5 | 2773 | 1.4 | 2793 | 1.4 | 2702 | 1.5 | 2943 |
| (50,5,4,3.0) | 3328 | 1.7 | 3297 | 1.7 | 3319 | 1.7 | 3213 | 1.8 | 3554 |
| (50,2,8,1.6) | 1186 | 1.7 | 1192 | 1.7 | 1198 | 1.7 | 1198 | 1.7 | 1204 |
| (50,2,8,2.2) | 3104 | 2.8 | 3107 | 2.8 | 3112 | 2.8 | 3101 | 2.9 | 3132 |
| (50,2,8,3.0) | 4166 | 2.8 | 4172 | 2.8 | 4197 | 2.8 | 4105 | 2.6 | 4233 |
| (50,3,8,1.6) | 5706 | 3.5 | 5711 | 3.5 | 5725 | 3.5 | 5710 | 3.6 | 5837 |
| (50,3,8,2.2) | 7141 | 4.6 | 7138 | 4.6 | 7150 | 4.5 | 7122 | 4.7 | 7296 |
| (50,3,8,3.0) | 8634 | 4.4 | 8601 | 4.4 | 8668 | 4.5 | 8413 | 4.2 | 8856 |
| (50,4,8,1.6) | 4761 | 2.7 | 4746 | 2.7 | 4764 | 2.7 | 4714 | 2.8 | 4912 |
| (50,4,8,2.2) | 6315 | 3.5 | 6304 | 3.5 | 6316 | 3.5 | 6196 | 3.6 | 6479 |
| (50,4,8,3.0) | 5437 | 3.2 | 5416 | 3.2 | 5441 | 3.2 | 5275 | 3.1 | 5589 |
| (50,5,8,1.6) | 6501 | 3.7 | 6496 | 3.7 | 6502 | 3.7 | 6408 | 3.8 | 6750 |
| (50,5,8,2.2) | 8594 | 5.1 | 8574 | 5.1 | 8605 | 5.1 | 8509 | 5.1 | 8899 |
| (50,5,8,3.0) | 6870 | 4.2 | 6839 | 4.2 | 6875 | 4.3 | 6516 | 3.8 | 7160 |

Table EC.7: Policy Performance (Mean and Standard Error (SE)) for Hub-and-Spoke Instances with 50 Periods Based on 100,000 Sample Paths

| Parameters | RB | | PB | | $\text{PB}_{\mathbf{c}}$ | | DEC | | Upp. |
|---|---|---|---|---|---|---|---|---|---|
| $(T,N,\kappa,\alpha)$ | Mean | SE | Mean | SE | Mean | SE | Mean | SE | Bnd. |
| (100,2,4,1.6) | 5667 | 1.5 | 5662 | 1.5 | 5677 | 1.5 | 5678 | 1.5 | 5739 |
| (100,2,4,2.2) | 9732 | 3.0 | 9725 | 3.0 | 9790 | 3.0 | 9827 | 3.1 | 9934 |
| (100,2,4,3.0) | 6955 | 3.0 | 6969 | 3.0 | 7062 | 2.9 | 7087 | 2.9 | 7217 |
| (100,3,4,1.6) | 5886 | 1.9 | 5876 | 1.8 | 5889 | 1.8 | 5863 | 1.9 | 6018 |
| (100,3,4,2.2) | 8468 | 2.5 | 8436 | 2.5 | 8478 | 2.5 | 8383 | 2.5 | 8642 |
| (100,3,4,3.0) | 4283 | 1.9 | 4280 | 1.9 | 4290 | 1.9 | 4243 | 1.9 | 4371 |
| (100,4,4,1.6) | 6320 | 2.3 | 6316 | 2.2 | 6331 | 2.2 | 6290 | 2.3 | 6488 |
| (100,4,4,2.2) | 5569 | 2.1 | 5570 | 2.1 | 5582 | 2.0 | 5538 | 2.1 | 5759 |
| (100,4,4,3.0) | 3511 | 1.6 | 3508 | 1.6 | 3516 | 1.6 | 3461 | 1.7 | 3637 |
| (100,5,4,1.6) | 5570 | 1.9 | 5567 | 1.9 | 5579 | 1.9 | 5518 | 2.0 | 5750 |
| (100,5,4,2.2) | 6417 | 2.4 | 6420 | 2.4 | 6416 | 2.4 | 6331 | 2.5 | 6692 |
| (100,5,4,3.0) | 3985 | 1.8 | 3974 | 1.7 | 3986 | 1.7 | 3941 | 1.8 | 4148 |
| (100,2,8,1.6) | 16081 | 5.0 | 16088 | 5.0 | 16109 | 5.0 | 16100 | 5.0 | 16202 |
| (100,2,8,2.2) | 14667 | 6.2 | 14673 | 6.2 | 14676 | 6.2 | 14647 | 6.3 | 14759 |
| (100,2,8,3.0) | 5834 | 2.3 | 5837 | 2.3 | 5843 | 2.2 | 5824 | 2.2 | 5907 |
| (100,3,8,1.6) | 13182 | 4.7 | 13185 | 4.6 | 13187 | 4.6 | 13112 | 4.8 | 13375 |
| (100,3,8,2.2) | 9331 | 4.8 | 9320 | 4.8 | 9342 | 4.8 | 9276 | 4.8 | 9437 |
| (100,3,8,3.0) | 6777 | 3.8 | 6785 | 3.8 | 6794 | 3.8 | 6770 | 3.8 | 6861 |
| (100,4,8,1.6) | 13921 | 5.4 | 13920 | 5.4 | 13937 | 5.3 | 13859 | 5.5 | 14218 |
| (100,4,8,2.2) | 10753 | 4.8 | 10747 | 4.7 | 10757 | 4.7 | 10679 | 4.9 | 10944 |
| (100,4,8,3.0) | 14702 | 6.0 | 14675 | 6.0 | 14693 | 6.0 | 14502 | 6.1 | 14992 |
| (100,5,8,1.6) | 13012 | 5.0 | 12984 | 4.9 | 13019 | 4.9 | 12932 | 5.0 | 13272 |
| (100,5,8,2.2) | 13665 | 5.4 | 13620 | 5.3 | 13649 | 5.3 | 13488 | 5.4 | 13954 |
| (100,5,8,3.0) | 8971 | 4.5 | 8958 | 4.4 | 8975 | 4.4 | 8847 | 4.6 | 9195 |

Table EC.8: Policy Performance (Mean and Standard Error (SE)) for Hub-and-Spoke Instances with 100 Periods Based on 100,000 Sample Paths

| Parameters | RB | | PB | | $\text{PB}_{\mathbf{c}}$ | | DEC | | Upp. |
|---|---|---|---|---|---|---|---|---|---|
| $(T, N, \kappa, \alpha)$ | Mean | SE | Mean | SE | Mean | SE | Mean | SE | Bnd. |
| (200,2,4,1.6) | 8541 | 2.1 | 8552 | 2.0 | 8590 | 2.0 | 8667 | 2.0 | 8742 |
| (200,2,4,2.2) | 16883 | 3.0 | 16883 | 3.0 | 17119 | 3.0 | 17098 | 3.1 | 17243 |
| (200,2,4,3.0) | 12737 | 2.9 | 12742 | 2.9 | 12898 | 2.7 | 12943 | 2.8 | 13094 |
| (200,3,4,1.6) | 10334 | 2.5 | 10344 | 2.5 | 10349 | 2.5 | 10345 | 2.5 | 10471 |
| (200,3,4,2.2) | 26204 | 5.4 | 26182 | 5.3 | 26330 | 5.2 | 26467 | 5.1 | 26900 |
| (200,3,4,3.0) | 11443 | 3.2 | 11450 | 3.1 | 11494 | 3.1 | 11551 | 3.1 | 11741 |
| (200,4,4,1.6) | 11937 | 2.7 | 11938 | 2.7 | 11943 | 2.6 | 11925 | 2.7 | 12156 |
| (200,4,4,2.2) | 8940 | 2.2 | 8919 | 2.2 | 8948 | 2.2 | 8930 | 2.2 | 9144 |
| (200,4,4,3.0) | 9672 | 2.7 | 9654 | 2.7 | 9669 | 2.7 | 9643 | 2.7 | 9894 |
| (200,5,4,1.6) | 13151 | 3.4 | 13152 | 3.3 | 13158 | 3.3 | 13164 | 3.5 | 13458 |
| (200,5,4,2.2) | 12250 | 2.8 | 12200 | 2.7 | 12226 | 2.7 | 12108 | 2.7 | 12545 |
| (200,5,4,3.0) | 13213 | 3.6 | 13187 | 3.6 | 13202 | 3.5 | 13139 | 3.6 | 13636 |
| (200,2,8,1.6) | 4786 | 3.2 | 4803 | 3.2 | 4809 | 3.2 | 4804 | 3.3 | 4831 |
| (200,2,8,2.2) | 25420 | 8.2 | 25445 | 8.1 | 25690 | 7.8 | 25728 | 7.9 | 25814 |
| (200,2,8,3.0) | 24296 | 9.0 | 24302 | 9.0 | 24488 | 8.7 | 24373 | 8.5 | 24565 |
| (200,3,8,1.6) | 23324 | 7.0 | 23336 | 7.0 | 23353 | 6.9 | 23348 | 7.0 | 23519 |
| (200,3,8,2.2) | 24427 | 6.7 | 24430 | 6.6 | 24555 | 6.5 | 24643 | 6.4 | 24847 |
| (200,3,8,3.0) | 17100 | 6.8 | 17110 | 6.8 | 17126 | 6.8 | 17094 | 6.8 | 17293 |
| (200,4,8,1.6) | 32237 | 8.0 | 32220 | 7.9 | 32239 | 7.9 | 32157 | 8.1 | 32638 |
| (200,4,8,2.2) | 29357 | 8.4 | 29342 | 8.3 | 29380 | 8.3 | 29376 | 8.3 | 29674 |
| (200,4,8,3.0) | 14460 | 4.6 | 14455 | 4.6 | 14480 | 4.6 | 14440 | 4.7 | 14701 |
| (200,5,8,1.6) | 22502 | 6.8 | 22489 | 6.8 | 22507 | 6.8 | 22455 | 6.8 | 22850 |
| (200,5,8,2.2) | 29924 | 9.0 | 29911 | 8.9 | 29936 | 8.9 | 29932 | 9.1 | 30343 |
| (200,5,8,3.0) | 14522 | 6.4 | 14522 | 6.3 | 14549 | 6.3 | 14486 | 6.4 | 14756 |

Table EC.9: Policy Performance (Mean and Standard Error (SE)) for Hub-and-Spoke Instances with 200 Periods Based on 100,000 Sample Paths

| Parameters | RB | | PB | | $\text{PB}_{\mathbf{c}}$ | | DEC | | Upp. |
|---|---|---|---|---|---|---|---|---|---|
| $(T, N, \kappa, \alpha)$ | Mean | SE | Mean | SE | Mean | SE | Mean | SE | Bnd. |
| (400,2,4,1.6) | 21155 | 3.5 | 21179 | 3.5 | 21657 | 3.5 | 22969 | 2.9 | 23052 |
| (400,2,4,2.2) | 15481 | 1.7 | 15462 | 1.7 | 15529 | 1.7 | 15516 | 1.7 | 15605 |
| (400,2,4,3.0) | 29057 | 5.2 | 29074 | 5.1 | 29558 | 4.9 | 29599 | 4.8 | 29749 |
| (400,3,4,1.6) | 53469 | 6.3 | 53427 | 6.3 | 53657 | 6.2 | 53946 | 6.1 | 54441 |
| (400,3,4,2.2) | 43910 | 5.8 | 43878 | 5.7 | 44054 | 5.8 | 44131 | 5.7 | 44616 |
| (400,3,4,3.0) | 22211 | 4.0 | 22202 | 3.9 | 22300 | 3.9 | 22381 | 3.8 | 22649 |
| (400,4,4,1.6) | 18708 | 3.4 | 18710 | 3.4 | 18717 | 3.4 | 18715 | 3.5 | 18930 |
| (400,4,4,2.2) | 28639 | 4.5 | 28607 | 4.5 | 28662 | 4.4 | 28647 | 4.5 | 29078 |
| (400,4,4,3.0) | 28893 | 5.6 | 28871 | 5.5 | 28938 | 5.5 | 28990 | 5.5 | 29432 |
| (400,5,4,1.6) | 23864 | 3.9 | 23843 | 3.8 | 23859 | 3.8 | 23856 | 3.8 | 24186 |
| (400,5,4,2.2) | 23685 | 4.0 | 23670 | 4.0 | 23693 | 4.0 | 23719 | 3.9 | 24129 |
| (400,5,4,3.0) | 14064 | 3.0 | 14048 | 2.9 | 14073 | 2.9 | 14049 | 2.9 | 14360 |
| (400,2,8,1.6) | 101108 | 14.9 | 101135 | 14.9 | 101680 | 15.1 | 102121 | 15.9 | 102319 |
| (400,2,8,2.2) | 49006 | 10.6 | 49071 | 10.6 | 50363 | 11.5 | 50464 | 11.7 | 50616 |
| (400,2,8,3.0) | 69646 | 11.8 | 69652 | 11.8 | 69911 | 11.6 | 69880 | 11.6 | 70119 |
| (400,3,8,1.6) | 55071 | 11.2 | 55104 | 11.1 | 55346 | 10.9 | 55781 | 10.5 | 56071 |
| (400,3,8,2.2) | 74023 | 13.4 | 74047 | 13.3 | 74618 | 13.1 | 74927 | 12.9 | 75401 |
| (400,3,8,3.0) | 66587 | 12.8 | 66571 | 12.7 | 66671 | 12.6 | 66679 | 12.7 | 67092 |
| (400,4,8,1.6) | 63338 | 11.0 | 63318 | 10.9 | 63433 | 10.9 | 63630 | 10.8 | 64099 |
| (400,4,8,2.2) | 50849 | 11.1 | 50864 | 11.0 | 51251 | 10.8 | 51623 | 10.8 | 51948 |
| (400,4,8,3.0) | 58200 | 13.3 | 58171 | 13.3 | 58318 | 13.1 | 58363 | 13.0 | 58915 |
| (400,5,8,1.6) | 34093 | 6.5 | 34077 | 6.4 | 34087 | 6.4 | 34034 | 6.5 | 34420 |
| (400,5,8,2.2) | 70634 | 12.2 | 70570 | 12.2 | 70756 | 12.1 | 70853 | 11.9 | 71586 |
| (400,5,8,3.0) | 76214 | 15.7 | 76172 | 15.6 | 76270 | 15.5 | 76291 | 15.3 | 77154 |

Table EC.10: Policy Performance (Mean and Standard Error (SE)) for Hub-and-Spoke Instances with 400 Periods Based on 100,000 Sample Paths

| | $T=50$ | | | | | $T=100$ | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Parameters | % Gain to RB | | % Gap to UB | | % $PB_c$ | % Gain to RB | | % Gap to UB | | % $PB_c$ |
| $(N,\kappa,\alpha)$ | PB | $PB_c$ | RB | $PB_c$ | To DEC | PB | $PB_c$ | RB | $PB_c$ | To DEC |
| (2,4,1.6) | 1.179 | 1.051 | 2.343 | 1.317 | -0.509 | -0.082 | 0.183 | 1.258 | 1.078 | -0.024 |
| (2,4,2.2) | 0.602 | 0.749 | 3.164 | 2.438 | -0.290 | -0.066 | 0.601 | 2.036 | 1.448 | -0.375 |
| (2,4,3.0) | 0.204 | 0.397 | 2.175 | 1.787 | 1.126 | 0.208 | 1.540 | 3.636 | 2.152 | -0.350 |
| (3,4,1.6) | -0.033 | 0.123 | 3.162 | 3.043 | 0.908 | -0.164 | 0.050 | 2.197 | 2.149 | 0.444 |
| (3,4,2.2) | -0.517 | 0.358 | 2.875 | 2.527 | 1.257 | -0.377 | 0.114 | 2.013 | 1.901 | 1.131 |
| (3,4,3.0) | -0.007 | 0.895 | 3.454 | 2.590 | 0.960 | -0.077 | 0.157 | 2.009 | 1.856 | 1.099 |
| (4,4,1.6) | -0.381 | 0.234 | 4.136 | 3.912 | 1.022 | -0.067 | 0.161 | 2.578 | 2.421 | 0.639 |
| (4,4,2.2) | -0.092 | -0.054 | 4.544 | 4.596 | 0.255 | 0.018 | 0.230 | 3.293 | 3.070 | 0.795 |
| (4,4,3.0) | -0.271 | 0.225 | 4.391 | 4.176 | 4.356 | -0.079 | 0.162 | 3.479 | 3.323 | 1.607 |
| (5,4,1.6) | -0.336 | 0.056 | 5.374 | 5.321 | 1.374 | -0.049 | 0.168 | 3.130 | 2.968 | 1.110 |
| (5,4,2.2) | -0.846 | -0.135 | 4.970 | 5.098 | 3.332 | 0.042 | -0.020 | 4.104 | 4.122 | 1.343 |
| (5,4,3.0) | -0.934 | -0.266 | 6.378 | 6.627 | 3.287 | -0.275 | 0.030 | 3.928 | 3.899 | 1.155 |
| (2,8,1.6) | 0.514 | 0.986 | 1.511 | 0.540 | 0.036 | 0.043 | 0.176 | 0.746 | 0.571 | 0.059 |
| (2,8,2.2) | 0.072 | 0.259 | 0.875 | 0.619 | 0.372 | 0.042 | 0.063 | 0.622 | 0.559 | 0.198 |
| (2,8,3.0) | 0.152 | 0.748 | 1.579 | 0.843 | 2.247 | 0.045 | 0.151 | 1.228 | 1.078 | 0.339 |
| (3,8,1.6) | 0.083 | 0.336 | 2.248 | 1.919 | 0.273 | 0.021 | 0.041 | 1.446 | 1.405 | 0.575 |
| (3,8,2.2) | -0.044 | 0.127 | 2.124 | 2.000 | 0.395 | -0.121 | 0.117 | 1.125 | 1.009 | 0.711 |
| (3,8,3.0) | -0.382 | 0.403 | 2.510 | 2.117 | 3.041 | 0.122 | 0.247 | 1.225 | 0.981 | 0.353 |
| (4,8,1.6) | -0.316 | 0.058 | 3.078 | 3.022 | 1.064 | -0.007 | 0.120 | 2.095 | 1.978 | 0.565 |
| (4,8,2.2) | -0.171 | 0.012 | 2.528 | 2.517 | 1.926 | -0.059 | 0.034 | 1.743 | 1.709 | 0.728 |
| (4,8,3.0) | -0.395 | 0.079 | 2.713 | 2.636 | 3.152 | -0.189 | -0.063 | 1.934 | 1.996 | 1.316 |
| (5,8,1.6) | -0.078 | 0.015 | 3.688 | 3.674 | 1.472 | -0.214 | 0.052 | 1.961 | 1.909 | 0.669 |
| (5,8,2.2) | -0.240 | 0.127 | 3.426 | 3.303 | 1.137 | -0.330 | -0.116 | 2.073 | 2.186 | 1.192 |
| (5,8,3.0) | -0.457 | 0.068 | 4.042 | 3.976 | 5.516 | -0.147 | 0.046 | 2.433 | 2.389 | 1.447 |

Table EC.11: Policy Comparison for Hub-and-Spoke Instances with 50 and 100 Periods Based on 100,000 Sample Paths

| | $T = 200$ | | | | | $T = 400$ | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Parameters | % Gain to RB | | % Gap to UB | | % $\text{PB}_\mathbf{c}$ | % Gain to RB | | % Gap to UB | | % $\text{PB}_\mathbf{c}$ |
| $(T, N, \kappa, \alpha)$ | PB | $\text{PB}_\mathbf{c}$ | RB | $\text{PB}_\mathbf{c}$ | To DEC | PB | $\text{PB}_\mathbf{c}$ | RB | $\text{PB}_\mathbf{c}$ | To DEC |
| (2,4,1.6) | 0.130 | 0.575 | 2.299 | 1.737 | -0.889 | 0.113 | 2.372 | 8.230 | 6.053 | -5.712 |
| (2,4,2.2) | 0.002 | 1.401 | 2.091 | 0.719 | 0.123 | -0.124 | 0.306 | 0.792 | 0.489 | 0.081 |
| (2,4,3.0) | 0.037 | 1.266 | 2.729 | 1.498 | -0.347 | 0.060 | 1.726 | 2.326 | 0.639 | -0.138 |
| (3,4,1.6) | 0.103 | 0.154 | 1.314 | 1.162 | 0.044 | -0.078 | 0.352 | 1.787 | 1.441 | -0.536 |
| (3,4,2.2) | -0.084 | 0.481 | 2.588 | 2.119 | -0.521 | -0.072 | 0.329 | 1.582 | 1.259 | -0.173 |
| (3,4,3.0) | 0.060 | 0.450 | 2.543 | 2.105 | -0.493 | -0.040 | 0.401 | 1.934 | 1.541 | -0.363 |
| (4,4,1.6) | 0.012 | 0.056 | 1.807 | 1.752 | 0.155 | 0.010 | 0.048 | 1.175 | 1.128 | 0.013 |
| (4,4,2.2) | -0.242 | 0.085 | 2.229 | 2.146 | 0.202 | -0.110 | 0.080 | 1.511 | 1.432 | 0.051 |
| (4,4,3.0) | -0.188 | -0.029 | 2.239 | 2.268 | 0.273 | -0.075 | 0.155 | 1.830 | 1.678 | -0.178 |
| (5,4,1.6) | 0.010 | 0.054 | 2.278 | 2.225 | -0.043 | -0.085 | -0.020 | 1.334 | 1.354 | 0.010 |
| (5,4,2.2) | -0.406 | -0.191 | 2.351 | 2.538 | 0.976 | -0.064 | 0.035 | 1.841 | 1.806 | -0.109 |
| (5,4,3.0) | -0.202 | -0.089 | 3.098 | 3.185 | 0.476 | -0.111 | 0.060 | 2.058 | 1.999 | 0.167 |
| (2,8,1.6) | 0.343 | 0.466 | 0.923 | 0.461 | 0.099 | 0.027 | 0.566 | 1.183 | 0.624 | -0.432 |
| (2,8,2.2) | 0.095 | 1.060 | 1.525 | 0.481 | -0.147 | 0.133 | 2.768 | 3.180 | 0.500 | -0.201 |
| (2,8,3.0) | 0.025 | 0.790 | 1.097 | 0.316 | 0.469 | 0.009 | 0.382 | 0.676 | 0.296 | 0.046 |
| (3,8,1.6) | 0.054 | 0.125 | 0.829 | 0.705 | 0.023 | 0.059 | 0.499 | 1.782 | 1.293 | -0.780 |
| (3,8,2.2) | 0.009 | 0.524 | 1.689 | 1.174 | -0.356 | 0.033 | 0.803 | 1.827 | 1.039 | -0.413 |
| (3,8,3.0) | 0.053 | 0.146 | 1.116 | 0.971 | 0.183 | -0.024 | 0.127 | 0.753 | 0.627 | -0.011 |
| (4,8,1.6) | -0.051 | 0.006 | 1.228 | 1.222 | 0.256 | -0.031 | 0.150 | 1.188 | 1.040 | -0.311 |
| (4,8,2.2) | -0.048 | 0.080 | 1.070 | 0.990 | 0.014 | 0.030 | 0.790 | 2.116 | 1.342 | -0.720 |
| (4,8,3.0) | -0.033 | 0.140 | 1.641 | 1.504 | 0.274 | -0.049 | 0.203 | 1.214 | 1.014 | -0.077 |
| (5,8,1.6) | -0.057 | 0.025 | 1.526 | 1.501 | 0.232 | -0.047 | -0.019 | 0.951 | 0.970 | 0.153 |
| (5,8,2.2) | -0.043 | 0.038 | 1.380 | 1.342 | 0.012 | -0.090 | 0.172 | 1.330 | 1.160 | -0.138 |
| (5,8,3.0) | -0.001 | 0.183 | 1.581 | 1.401 | 0.431 | -0.055 | 0.074 | 1.219 | 1.146 | -0.027 |

Table EC.12: Policy Comparison for Hub-and-Spoke Instances with 200 and 400 Periods Based on 100,000 Sample Paths

## EC.3. Detailed Results on Computational Performance

For all tables in Sections EC.3.1 and EC.3.2, the first column gives the parameter $(I, \kappa, \alpha)$ for each hotel instance or $(N, \kappa, \alpha)$ for each hub-and-spoke instance. The remaining columns show the solution times for $T = 50, 100, 200$, and 400 with various formulations, respectively. A number labeled with * means that Gurobi terminates with "Suboptimal" status, which means that Gurobi is "unable to satisfy optimality tolerances; a suboptimal solution is available"(Gurobi Optimization 2022). Otherwise, Gurobi terminates with "Optimal" status. Similarly, we report the solution times for the dynamic programming decomposition (DEC) in Section EC.3.3.

### EC.3.1. Testing (R), (P), and ($D_c$) with `BarTol=1e-6`

| | $T=50$ | | | $T=100$ | | | $T=200$ | | | $T=400$ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $(I,\kappa,\alpha)$ | (R) | (P) | ($D_c$) | (R) | (P) | ($D_c$) | (R) | (P) | ($D_c$) | (R) | (P) | ($D_c$) |
| (2,4,1.6) | 0.7 | 0.4 | 0.3 | 2.1 | 1.2 | 1.0 | 7.6 | 5.3 | 4.4 | 47.7 | 26.7 | 39.4 |
| (2,4,2.2) | 0.3 | 0.2 | 0.3 | 2.1 | 1.4 | 2.5 | 9.5 | 6.7 | 6.6 | 48.7 | 34.2 | 27.8 |
| (2,4,3.0) | 0.2 | 0.2 | 0.1 | 0.8 | 0.8 | 0.8 | 3.9 | 2.5 | 2.6 | 17.3 | 11.7 | 13.3 |
| (3,4,1.6) | 0.7 | 0.5 | 0.3 | 3.8 | 3.6 | 1.9 | 15.7 | 15.6 | 11.8 | 73.4 | 73.2 | 63.1 |
| (3,4,2.2) | 0.7 | 0.7 | 0.4 | 2.1 | 2.2 | 1.5 | 18.0 | 18.3 | 11.5 | 123.4 | 96.0 | 82.7 |
| (3,4,3.0) | 0.6 | 0.4 | 0.3 | 2.3 | 1.9 | 1.5 | 13.6 | 14.3 | 8.1 | 38.7 | 36.3 | 33.7 |
| (4,4,1.6) | 1.2 | 0.9 | 0.8 | 7.7 | 5.4 | 4.7 | 45.1 | 34.4 | 28.4 | 196.1 | 160.1 | 152.2 |
| (4,4,2.2) | 1.4 | 0.9 | 0.9 | 4.9 | 3.8 | 3.0 | 19.1 | 16.9 | 15.4 | 289.3 | 253.7 | 184.8 |
| (4,4,3.0) | 1.1 | 0.7 | 0.7 | 4.2 | 3.1 | 2.9 | 18.0 | 16.9 | 12.1 | 110.2 | 127.1 | 73.9 |
| (5,4,1.6) | 3.8 | 2.1 | 1.7 | 14.7 | 12.9 | 9.0 | 69.3 | 74.6 | 48.1 | 502.0 | 432.0 | 263.0 |
| (5,4,2.2) | 2.7 | 1.9 | 1.5 | 10.6 | 11.2 | 5.9 | 115.1 | 68.3 | 62.7 | 461.6 | 322.2 | 418.4 |
| (5,4,3.0) | 1.4 | 0.8 | 1.0 | 9.4 | 5.7 | 4.0 | 45.5 | 45.7 | 26.3 | 243.2 | 128.3 | 126.6 |
| (2,8,1.6) | 0.4 | 0.3 | 0.3 | 1.8 | 1.2 | 1.1 | 8.5 | 4.8 | 5.2 | 52.2 | 27.8 | 36.9 |
| (2,8,2.2) | 0.3 | 0.3 | 0.2 | 1.5 | 0.9 | 1.0 | 8.6 | 5.5 | 5.1 | 50.0 | 25.2 | 27.2 |
| (2,8,3.0) | 0.2 | 0.2 | 0.3 | 0.8 | 0.5 | 0.8 | 4.1 | 2.2 | 3.1 | 24.5 | 15.7 | 16.0 |
| (3,8,1.6) | 1.0 | 0.8 | 0.5 | 4.8 | 3.4 | 2.7 | 18.9 | 18.8 | 14.2 | 69.4 | 45.8 | 56.5 |
| (3,8,2.2) | 0.7 | 0.6 | 0.4 | 2.3 | 2.2 | 1.8 | 24.8 | 25.1 | 14.5 | 120.1 | 76.2 | 75.2 |
| (3,8,3.0) | 0.5 | 0.5 | 0.3 | 2.4 | 1.8 | 1.6 | 8.2 | 9.2 | 7.9 | 22.7 | 26.8 | 20.8 |
| (4,8,1.6) | 1.6 | 1.1 | 1.1 | 9.4 | 8.4 | 6.1 | 43.0 | 29.4 | 30.4 | 133.7 | 114.4 | 121.7 |
| (4,8,2.2) | 1.7 | 1.0 | 1.2 | 6.0 | 4.3 | 4.0 | 43.8 | 37.8 | 26.1 | ***500.1** | 246.0 | 205.4 |
| (4,8,3.0) | 0.8 | 0.6 | 0.6 | 2.9 | 2.3 | 2.0 | 17.0 | 13.3 | 16.5 | 81.6 | 73.5 | 75.7 |
| (5,8,1.6) | 3.9 | 3.1 | 2.0 | 15.4 | 11.4 | 8.1 | 85.4 | 83.9 | 58.5 | 574.1 | 767.5 | 409.9 |
| (5,8,2.2) | 2.0 | 1.2 | 1.1 | 13.5 | 12.6 | 7.6 | 98.8 | 108.5 | 72.0 | 458.4 | 309.9 | 371.8 |
| (5,8,3.0) | 1.5 | 1.1 | 1.0 | 9.4 | 8.1 | 4.5 | 43.9 | 46.0 | 25.7 | ***214.0** | 245.8 | 139.4 |

Table EC.13: Computational Performance on the Hotel Instances with BarTol = 1e-6. The Results Are Numbers of Seconds for Gurobi to Terminate. * Indicates that Gurobi Terminates With "Suboptimal" Status.

| | $T=50$ | | | $T=100$ | | | $T=200$ | | | $T=400$ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $(N,\kappa,\alpha)$ | (R) | (P) | ($\mathrm{D}_{\mathbf{c}}$) | (R) | (P) | ($\mathrm{D}_{\mathbf{c}}$) | (R) | (P) | ($\mathrm{D}_{\mathbf{c}}$) | (R) | (P) | ($\mathrm{D}_{\mathbf{c}}$) |
| (2,4,1.6) | 0.4 | 0.3 | 0.2 | 1.5 | 1.0 | 0.9 | 7.2 | 5.1 | 5.2 | 43.5 | 30.7 | 27.8 |
| (2,4,2.2) | 0.3 | 0.1 | 0.2 | 1.2 | 0.7 | 0.8 | 5.3 | 4.1 | 3.4 | 24.6 | 14.6 | 14.5 |
| (2,4,3.0) | 0.1 | 0.1 | 0.1 | 0.9 | 0.6 | 0.6 | 5.0 | 2.8 | 3.6 | 31.2 | 36.8 | 20.0 |
| (3,4,1.6) | 0.5 | 0.3 | 0.3 | 2.2 | 2.3 | 1.7 | 13.5 | 15.0 | 8.1 | 52.6 | 48.2 | 44.0 |
| (3,4,2.2) | 0.6 | 0.4 | 0.3 | 1.6 | 1.2 | 1.0 | 8.0 | 9.2 | 5.1 | 34.2 | 43.2 | 30.6 |
| (3,4,3.0) | 0.4 | 0.3 | 0.3 | 0.9 | 0.8 | 0.8 | 7.5 | 8.7 | 5.0 | 30.5 | 26.2 | 21.9 |
| (4,4,1.6) | 0.8 | 0.5 | 0.7 | 3.6 | 2.8 | 2.6 | 20.8 | 15.8 | 15.3 | 76.6 | 60.0 | 64.0 |
| (4,4,2.2) | 0.7 | 0.5 | 0.7 | 3.0 | 2.0 | 2.0 | 16.3 | 12.6 | 12.7 | 49.8 | 40.1 | 47.0 |
| (4,4,3.0) | 0.5 | 0.3 | 0.5 | 2.6 | 2.0 | 1.7 | 9.1 | 5.9 | 5.8 | 54.9 | 37.0 | 38.7 |
| (5,4,1.6) | 1.1 | 0.7 | 1.1 | 6.3 | 4.6 | 4.4 | 29.5 | 29.2 | 25.8 | 143.4 | 111.3 | 112.1 |
| (5,4,2.2) | 0.8 | 0.5 | 0.8 | 4.5 | 3.1 | 3.3 | 19.1 | 13.7 | 16.1 | 97.7 | 94.0 | 73.3 |
| (5,4,3.0) | 0.7 | 0.4 | 1.0 | 3.4 | 2.4 | 3.0 | 11.9 | 8.7 | 10.5 | 47.9 | 43.1 | 45.2 |
| (2,8,1.6) | 0.3 | 0.2 | 0.2 | 1.7 | 1.2 | 1.1 | 7.6 | 4.6 | 4.8 | 49.5 | 23.0 | 30.6 |
| (2,8,2.2) | 0.2 | 0.1 | 0.2 | 1.0 | 0.6 | 0.7 | 8.9 | 5.1 | 4.5 | 26.8 | 21.2 | 22.4 |
| (2,8,3.0) | 0.1 | 0.1 | 0.2 | 1.0 | 0.5 | 0.6 | 6.9 | 4.9 | 4.8 | 19.0 | 12.9 | 13.9 |
| (3,8,1.6) | 0.6 | 0.4 | 0.3 | 2.5 | 2.8 | 1.9 | 8.8 | 11.1 | 6.6 | ***118.6** | 106.8 | 80.4 |
| (3,8,2.2) | 0.5 | 0.6 | 0.3 | 1.3 | 1.3 | 1.3 | 11.8 | 12.1 | 6.2 | 43.5 | 53.5 | 34.5 |
| (3,8,3.0) | 0.2 | 0.1 | 0.2 | 1.5 | 1.0 | 0.9 | 5.8 | 5.3 | 4.5 | 46.6 | 52.7 | 27.3 |
| (4,8,1.6) | 0.7 | 0.5 | 0.6 | 5.0 | 3.5 | 3.0 | 16.2 | 12.8 | 14.4 | 109.2 | 96.3 | 106.1 |
| (4,8,2.2) | 0.5 | 0.4 | 0.6 | 2.5 | 1.5 | 1.8 | 16.7 | 15.1 | 12.7 | 82.4 | 82.7 | 68.8 |
| (4,8,3.0) | 0.5 | 0.2 | 0.4 | 1.8 | 1.1 | 1.5 | 8.0 | 5.9 | 6.8 | 45.4 | 41.4 | 29.7 |
| (5,8,1.6) | 1.1 | 0.6 | 1.0 | 5.7 | 4.8 | 5.0 | 29.4 | 26.6 | 22.2 | 117.2 | 94.2 | 133.2 |
| (5,8,2.2) | 1.0 | 0.6 | 1.1 | 5.7 | 4.6 | 3.8 | 28.0 | 18.2 | 16.8 | 115.5 | 91.4 | 94.4 |
| (5,8,3.0) | 0.7 | 0.4 | 0.7 | 3.7 | 2.1 | 2.8 | 19.0 | 10.3 | 11.5 | 81.5 | 77.7 | 59.2 |

Table EC.14: Computational Performance on the Hub-and-Spoke Instances with BarTol = 1e-6. The Results Are Numbers of Seconds for Gurobi to Terminate. * Indicates that Gurobi Terminates With "Suboptimal" Status.

### EC.3.2. Testing (P) and ($\mathbf{D_c}$) with `BarTol=1e-7`

| | $T=50$ | | $T=100$ | | $T=200$ | | $T=400$ | |
|---|---|---|---|---|---|---|---|---|
| $(I,\kappa,\alpha)$ | (P) | ($\mathrm{D}_\mathbf{c}$) | (P) | ($\mathrm{D}_\mathbf{c}$) | (P) | ($\mathrm{D}_\mathbf{c}$) | (P) | ($\mathrm{D}_\mathbf{c}$) |
| (2,4,1.6) | 0.4 | 0.4 | 1.4 | 1.3 | 6.9 | 6.0 | 49.3 | 49.9 |
| (2,4,2.2) | 0.2 | 0.3 | 2.4 | 2.8 | 8.2 | 9.8 | 61.0 | 47.8 |
| (2,4,3.0) | 0.2 | 0.1 | 0.9 | 0.8 | 3.1 | 3.5 | ***17.0** | 16.0 |
| (3,4,1.6) | 0.6 | 0.3 | 3.9 | 2.5 | 16.4 | 15.8 | ***85.3** | 148.6 |
| (3,4,2.2) | 0.8 | 0.5 | 2.4 | 1.9 | 19.6 | 19.8 | 158.7 | 273.1 |
| (3,4,3.0) | 0.4 | 0.3 | 2.3 | 1.6 | 15.1 | 11.8 | 41.8 | 48.2 |
| (4,4,1.6) | 1.0 | 0.8 | 5.7 | 5.7 | 61.2 | 61.8 | ***183.1** | 435.8 |
| (4,4,2.2) | 1.0 | 1.1 | 4.1 | 4.4 | 20.8 | 19.2 | ***325.8** | 513.4 |
| (4,4,3.0) | 0.8 | 0.8 | 4.0 | 3.4 | 17.8 | 14.8 | ***134.1** | 146.6 |
| (5,4,1.6) | 2.4 | 2.3 | 13.1 | 14.1 | 96.9 | 80.7 | ***466.5** | 391.7 |
| (5,4,2.2) | 2.2 | 1.8 | 11.3 | 7.9 | 96.1 | 211.9 | ***349.7** | 537.9 |
| (5,4,3.0) | 1.0 | 1.1 | 6.2 | 5.6 | 45.6 | 32.8 | 170.7 | 305.9 |
| (2,8,1.6) | 0.3 | 0.3 | 1.7 | 1.5 | 8.6 | 6.3 | 36.4 | 54.1 |
| (2,8,2.2) | 0.3 | 0.3 | 1.1 | 1.2 | 7.5 | 7.2 | 53.9 | 54.8 |
| (2,8,3.0) | 0.2 | 0.3 | 0.6 | 1.0 | 2.3 | 3.5 | 20.1 | 20.9 |
| (3,8,1.6) | 0.9 | 0.7 | 3.9 | 3.2 | 26.0 | 29.0 | ***72.1** | 126.8 |
| (3,8,2.2) | 0.6 | 0.4 | 2.4 | 2.1 | 25.5 | 37.5 | 160.1 | 279.1 |
| (3,8,3.0) | 0.5 | 0.3 | 2.3 | 1.9 | 9.8 | 9.7 | 37.9 | 47.3 |
| (4,8,1.6) | 1.6 | 1.5 | 9.7 | 7.1 | 61.3 | 72.6 | ***128.1** | 282.7 |
| (4,8,2.2) | 1.1 | 1.3 | 4.3 | 5.5 | 36.5 | 77.8 | 481.1 | ***297.0** |
| (4,8,3.0) | 0.7 | 0.7 | 2.7 | 2.6 | 13.6 | 19.8 | ***87.3** | 143.3 |
| (5,8,1.6) | 3.6 | 2.1 | 10.7 | 9.1 | 103.3 | 203.7 | ***829.7** | 617.1 |
| (5,8,2.2) | 1.6 | 1.3 | 10.2 | 7.6 | 125.5 | ***100.5** | ***357.5** | ***388.4** |
| (5,8,3.0) | 1.2 | 1.0 | 7.2 | 5.6 | 48.3 | ***35.6** | ***281.9** | 276.1 |

Table EC.15: Computational Performance on the Hotel Instances with BarTol = 1e-7. The Results Are Numbers of Seconds for Gurobi to Terminate. * Indicates that Gurobi Terminates With "Sub-optimal" Status.

| | $T=50$ | | $T=100$ | | $T=200$ | | $T=400$ | |
|---|---|---|---|---|---|---|---|---|
| $(N,\kappa,\alpha)$ | (P) | ($\mathrm{D}_{\mathbf{c}}$) | (P) | ($\mathrm{D}_{\mathbf{c}}$) | (P) | ($\mathrm{D}_{\mathbf{c}}$) | (P) | ($\mathrm{D}_{\mathbf{c}}$) |
| (2,4,1.6) | 0.3 | 0.3 | 1.3 | 1.4 | 5.3 | 7.2 | ***40.1** | 112.3 |
| (2,4,2.2) | 0.2 | 0.2 | 0.9 | 1.2 | 4.6 | 4.0 | 19.9 | 20.5 |
| (2,4,3.0) | 0.1 | 0.1 | 0.8 | 2.6 | 7.5 | 3.5 | 32.9 | 47.5 |
| (3,4,1.6) | 0.7 | 0.4 | 3.0 | 2.3 | 16.3 | 8.5 | ***67.7** | 69.7 |
| (3,4,2.2) | 0.5 | 0.4 | 1.8 | 1.9 | 9.1 | 7.0 | 45.9 | 55.1 |
| (3,4,3.0) | 0.2 | 0.2 | 1.0 | 1.2 | 8.6 | 6.2 | 31.1 | 29.9 |
| (4,4,1.6) | 0.8 | 0.8 | 3.1 | 2.9 | 24.9 | 16.7 | 137.9 | 241.1 |
| (4,4,2.2) | 0.7 | 0.9 | 2.3 | 2.5 | 12.3 | 12.6 | ***53.6** | 119.3 |
| (4,4,3.0) | 0.3 | 0.7 | 2.2 | 1.9 | 6.3 | 6.6 | ***40.5** | 41.8 |
| (5,4,1.6) | 1.0 | 1.2 | 5.5 | 4.4 | 29.9 | 27.4 | 134.0 | 195.1 |
| (5,4,2.2) | 0.6 | 0.8 | 3.5 | 3.4 | 20.1 | 18.6 | 115.4 | 199.4 |
| (5,4,3.0) | 0.4 | 0.8 | 2.6 | 3.2 | 9.0 | 19.3 | 87.5 | 94.6 |
| (2,8,1.6) | 0.3 | 0.2 | 1.4 | 1.3 | 4.5 | 5.1 | 34.0 | 67.2 |
| (2,8,2.2) | 0.2 | 0.2 | 0.7 | 0.9 | 7.2 | 6.2 | 30.5 | 34.6 |
| (2,8,3.0) | 0.1 | 0.1 | 0.7 | 0.7 | 5.3 | 5.2 | 15.5 | 15.6 |
| (3,8,1.6) | 0.5 | 0.5 | 3.0 | 1.9 | 11.1 | 8.9 | ***117.8** | 152.1 |
| (3,8,2.2) | 0.7 | 0.5 | 1.4 | 1.5 | 12.3 | 8.9 | 57.4 | 52.9 |
| (3,8,3.0) | 0.2 | 0.2 | 1.1 | 1.2 | 5.1 | 5.3 | 55.8 | 68.8 |
| (4,8,1.6) | 0.7 | 0.8 | 4.0 | 3.6 | ***16.0** | 16.2 | ***110.2** | 307.5 |
| (4,8,2.2) | 0.4 | 0.7 | 1.8 | 2.0 | ***14.9** | ***18.9** | ***95.1** | 200.5 |
| (4,8,3.0) | 0.3 | 0.5 | 1.2 | 1.7 | 6.4 | 9.7 | ***48.7** | 57.5 |
| (5,8,1.6) | 0.7 | 1.1 | 4.9 | 5.5 | ***32.3** | 42.5 | 135.6 | 297.7 |
| (5,8,2.2) | 0.8 | 1.3 | 4.4 | 4.2 | 20.0 | 19.7 | ***94.1** | 212.7 |
| (5,8,3.0) | 0.5 | 0.8 | 2.2 | 3.1 | 11.3 | 13.0 | 97.7 | 112.2 |

Table EC.16: Computational Performance on the Hub-and-Spoke Instances with BarTol = 1e-7. The Results Are Numbers of Seconds for Gurobi to Terminate. * Indicates that Gurobi Terminates With “Suboptimal” Status.

### EC.3.3. Running Time of the Dynamic Programming Decomposition (DEC)

| Hotel Instances | | | | | Hub-and-Spoke Instances | | | | |
|---|---|---|---|---|---|---|---|---|---|
| $(I,\kappa,\alpha)$ | $T=50$ | $T=100$ | $T=200$ | $T=400$ | $(N,\kappa,\alpha)$ | $T=50$ | $T=100$ | $T=200$ | $T=400$ |
| (2,4,1.6) | 0.1 | 0.5 | 2.0 | 8.4 | (2,4,1.6) | 0.3 | 0.9 | 4.2 | 18.1 |
| (2,4,2.2) | 0.1 | 0.4 | 1.6 | 6.1 | (2,4,2.2) | 0.2 | 0.8 | 3.1 | 11.5 |
| (2,4,3.0) | 0.1 | 0.2 | 1.1 | 4.6 | (2,4,3.0) | 0.1 | 0.6 | 2.6 | 9.0 |
| (3,4,1.6) | 0.2 | 0.9 | 5.0 | 16.7 | (3,4,1.6) | 0.6 | 2.5 | 9.4 | 46.9 |
| (3,4,2.2) | 0.2 | 0.7 | 3.5 | 14.9 | (3,4,2.2) | 0.5 | 1.7 | 8.2 | 32.7 |
| (3,4,3.0) | 0.1 | 0.6 | 2.3 | 10.8 | (3,4,3.0) | 0.4 | 1.2 | 5.9 | 22.8 |
| (4,4,1.6) | 0.4 | 2.2 | 8.7 | 34.9 | (4,4,1.6) | 1.3 | 4.8 | 21.1 | 78.1 |
| (4,4,2.2) | 0.4 | 1.6 | 6.6 | 31.6 | (4,4,2.2) | 1.1 | 3.7 | 15.4 | 60.0 |
| (4,4,3.0) | 0.3 | 1.2 | 4.1 | 20.3 | (4,4,3.0) | 0.7 | 2.9 | 10.9 | 45.0 |
| (5,4,1.6) | 0.9 | 4.6 | 14.9 | 65.5 | (5,4,1.6) | 2.6 | 9.0 | 37.4 | 136.8 |
| (5,4,2.2) | 0.8 | 3.0 | 15.9 | 65.7 | (5,4,2.2) | 1.6 | 6.6 | 27.2 | 109.1 |
| (5,4,3.0) | 0.6 | 1.9 | 9.6 | 35.7 | (5,4,3.0) | 1.5 | 5.1 | 21.0 | 76.5 |
| (2,8,1.6) | 0.1 | 0.5 | 1.8 | 9.1 | (2,8,1.6) | 0.2 | 1.1 | 3.2 | 17.5 |
| (2,8,2.2) | 0.1 | 0.4 | 1.6 | 5.9 | (2,8,2.2) | 0.2 | 0.6 | 3.3 | 14.3 |
| (2,8,3.0) | 0.1 | 0.3 | 1.0 | 4.7 | (2,8,3.0) | 0.1 | 0.6 | 2.7 | 10.0 |
| (3,8,1.6) | 0.2 | 1.2 | 4.2 | 18.3 | (3,8,1.6) | 0.6 | 2.4 | 9.8 | 46.3 |
| (3,8,2.2) | 0.2 | 0.7 | 3.8 | 15.0 | (3,8,2.2) | 0.5 | 1.6 | 7.7 | 34.3 |
| (3,8,3.0) | 0.1 | 0.5 | 2.5 | 7.0 | (3,8,3.0) | 0.3 | 1.2 | 5.4 | 24.8 |
| (4,8,1.6) | 0.6 | 2.1 | 9.1 | 31.7 | (4,8,1.6) | 1.3 | 5.3 | 20.8 | 88.5 |
| (4,8,2.2) | 0.4 | 1.6 | 6.9 | 32.9 | (4,8,2.2) | 1.0 | 3.7 | 16.0 | 66.8 |
| (4,8,3.0) | 0.3 | 1.0 | 5.0 | 17.8 | (4,8,3.0) | 0.8 | 2.6 | 11.0 | 43.7 |
| (5,8,1.6) | 0.9 | 3.9 | 16.3 | 72.6 | (5,8,1.6) | 2.5 | 9.0 | 37.0 | 145.8 |
| (5,8,2.2) | 0.7 | 2.9 | 15.9 | 50.8 | (5,8,2.2) | 1.9 | 6.9 | 28.6 | 115.1 |
| (5,8,3.0) | 0.5 | 2.1 | 10.1 | 33.6 | (5,8,3.0) | 1.4 | 5.2 | 20.2 | 85.2 |

Table EC.17: Running Time in Seconds of the Dynamic Programming Decomposition (DEC)