

# Dynamic Pricing for Network Revenue Management: A New Approach and Application in the Hotel Industry

Dan Zhang

Leeds School of Business, University of Colorado at Boulder  
dan.zhang@colorado.edu

Larry Weatherford

College of Business, University of Wyoming  
lrw@uwyo.edu

Dynamic pricing for network revenue management has received considerable attention in research and practice. Based on data obtained from a major hotel, we use a large-scale numerical study to compare the performance of several heuristic approaches proposed in the literature. The heuristic approaches we consider include deterministic linear programming with resolving and three variants of dynamic programming decomposition. Dynamic programming decomposition is considered one of the strongest heuristics and is the method chosen in some recent commercial implementations, and remains a topic of research in the recent academic literature. In addition to a plain-vanilla implementation of dynamic programming decomposition, we consider the variants proposed by Erdelyi and Topaloglu (2011) and Zhang (2011). For the base scenario generated from the real data, we show that the method based on Zhang (2011) leads to a small but significant lift in revenue compared with all other approaches. We generate many alternative problem scenarios by varying capacity-demand ratio and network structure and show that the performance of the different heuristics can be strongly influenced by both. Overall, our paper shows the promise of some recent proposals in the academic literature but also offers a cautionary tale on the choice of heuristic methods for practical network pricing problems.

*Key words:* Revenue management; dynamic pricing; approximate dynamic programming

*This version:* March 21, 2016

---

## 1. Introduction and Literature Review

Two popular approaches to solving the network revenue management (RM) problem are deterministic linear programming (DLP) and dynamic programming (DP). DLP, when frequently resolved, generates good control policies. DLP has been shown to generate greater expected revenues of up to 2.9% over traditional leg-based RM methods like Expected Marginal Seat Revenue (EMSR) by dealing explicitly with the varying lengths-of-stay by guests (Weatherford 1995). The main weakness of DLP is that it systematically ignores demand uncertainty by only taking into account expected demand. This weakness can be remedied by a stochastic DP formulation. However, DP's state space can easily suffer from Bellman's curse of dimensionality. It is common in practice to

---

decompose the network problem into a collection of smaller ones, where each involves only one resource (i.e., a single leg or single stay night). Such an approach (called leg DP in practice or DCOMP in this paper) has been widely applied in the airline industry and has also been adopted by the hotel industry in recent years; a nice introduction to DCOMP can be found in Talluri and van Ryzin (2004), Chapter 4.

Hotel revenue management can be cast as network revenue management by treating each room-night as a separate resource; see, e.g., Gallego and van Ryzin (1997). Multi-day stays are then analogous to multi-leg itineraries in an airline network. Unlike airline RM, hotel RM problems do not have a clear end of horizon.<sup>1</sup> Typically, rolling-horizon procedures are used to solve the problem at a given cut-off date.<sup>2</sup> Such a formulation is indeed considered by many practical RM systems. Compared with an airline network RM problem, network effects can be even more pronounced. In the popular hub-and-spoke network in the airline industry, each customer generally travels at most two legs, while it is not uncommon for a customer to stay for a week at a hotel, equivalent to a seven-leg itinerary in an airline network.

There is very limited published research in hotel revenue management that applies to actual industry data (Bodea et al. 2009). We apply the approach developed in this paper to data obtained from a major hotel. The data consists of stays of up to 12 consecutive nights over a 30 day arrival horizon from mid-July to mid-August. The resulting problem is equivalent to a flight network with 30 legs and itineraries that use up to 12 legs. The problem size we consider in this paper is the largest that we are aware of that has been reported in the literature

DCOMP is considered one of the strongest heuristics for network revenue management in academic research and in practice; see, for example, Talluri and van Ryzin (2004), Liu and van Ryzin (2008), Zhang and Adelman (2009) and Vulcano et al. (2010). Because of its strong performance in numerical tests, DCOMP remains an actively researched topic. Two recent papers that consider variants of DCOMP are Erdelyi and Topaloglu (2011) and Zhang (2011). Erdelyi and Topaloglu (2011) propose to approximate the marginal cost of accepting a customer to be the average marginal cost of the resources used by the customer (which we call DCOMP-AVG). They show that the approach works very well on two simulated data sets. Zhang (2011) proposes an improvement to DCOMP (which we call DCOMP1) by considering a non-linear non-separable functional approximation to dynamic programming value functions. Since the functional approximation

<sup>1</sup> When round-trips are considered, the end of horizon is also not clearly defined for airline RM problems. However, standard industry practice is to map round-trip demand to one-way demand and then solve one-way problems with the departure date considered as the end of horizon.

<sup>2</sup> We point out that even though an infinite horizon formulation is possible, such a formulation would require non-stationary parameters over time and pose practical solution challenges.

is non-separable, it retains at least partial network information in the solution process. Indeed, the solution approach involves a *simultaneous dynamic programming* approach where the single-resource DPs are solved simultaneously over time and exchange intermediate values in between. This should be contrasted with DCOMP where each single-resource DP is solved independently of all other single-resource DPs. He shows that the approach leads to better revenue bounds compared with DCOMP. A numerical study based on simulated airline data of a four-leg, hub-and-spoke network showed that the approach led to better network RM control policies, boosting expected revenue by an average of 1.07% over DCOMP across the 20 cases tested (with gains up to 4.4%). More importantly, his approach does not require a significant increase in computational time compared with DCOMP.

The work of Zhang (2011) is based on LP-based approximate dynamic programming. A formal framework with applications to network revenue management is introduced by Adelman (2007). In the last several years, there has been significant research effort devoted to this approach. Much of the existing work deals with network revenue management with independent demand assumption (Farias and Van Roy 2007, Kunnumkal and Topaloglu 2010a) and more recently models with discrete choice behavior (Zhang and Adelman 2009, Kunnumkal and Topaloglu 2008, Meissner et al. 2012). Research adopting the framework for dynamic pricing problems is rather limited. Zhang and Lu (2013) apply the framework to network pricing problems. They point out a number of potential complexities involved. In particular, the equivalent linear programming formulation to dynamic program is a semi-infinite linear program. They also generalize the network decomposition approach to network pricing problem, which involves solving a deterministic formulation which is a nonlinear program instead of a linear program. Our model formulation follows Zhang and Lu (2013). We show that applying a nonlinear non-separable functional approximation leads to an improved bound compared with the network decomposition in Zhang and Lu (2013). This result is a generalization of the results in Zhang (2011).

Using the data obtained from the hotel, we compare the performance of five different approaches: the deterministic linear programming approach with and without resolving (which we call DLP16 and DLP), DCOMP, DCOMP-AVG, and DCOMP1. For the base-case scenario, we show that DCOMP1 produces the highest average revenue in our simulation study. Even though the relative magnitude of the revenue improvement is small, it is quite substantial in practical terms (e.g., a 0.1% revenue gain for an airline/hotel company with annual revenues of \$10 billion would be \$10 million).

In order to fully understand the drivers of numerical performance for the different heuristic methods, we generate many different problem scenarios by varying demand/capacity ratios and

---

network structures. While it is fairly standard to consider different demand/capacity ratios in numerical studies, we are not aware of any paper that considers problem instances with different network structures. For the data set that we obtained, customers can stay up to 12 nights, equivalent to a 12-leg itinerary in airline networks. This implies that hotel networks have much stronger *network effects* compared with airline networks. We are therefore interested in understanding how the much more complex hotel networks affect the performance of the different heuristics. We do this by mapping the problem to test instances with progressively shorter maximum length of stay for all demand/capacity ratios. In our numerical study, we consider problem instances with maximum lengths of stay of 12, 9, 6, and 2 nights.

Our numerical study confirms that the network structure has profound impact on the performance of the different heuristics. The heuristic based on deterministic linear programming with resolving emerged as a very strong heuristic when the maximum lengths of stay are 12, 9, and 6 nights. When the maximum length of stay is reduced to 2, it is dominated by DCOMP1, even though the latter does not resolve. A key insight of our research is the importance of considering network structure when choosing the heuristic control method. DLP and DCOMP offer very different trade-offs. DLP formulation captures the network effects well as it fully incorporates the product consumption matrix information, but it does so at the expense of ignoring demand uncertainty. DCOMP, on the other hand, fully incorporates demand uncertainty; however, it only captures the network effects through the bid-prices and fare proration steps. DCOMP-AVG and DCOMP1 attempt to remedy the deficiencies of DCOMP. Our numerical results suggest that their performance are scenario dependent. Practitioners are therefore urged to consider the various trade-offs when selecting the method for their particular applications, as in general, no method is the clear winner in all problem instances. A simulation study like the one carried out in this paper is a reasonable way to do that.

The remainder of the paper is organized as follows. We briefly review the relevant literature before moving on to model formulation in Section 2. Section 3 introduces several alternative solution strategies. Section 4 reports data from a case study at a major hotel. Section 5 describes control policies from the four alternative solution strategies. Section 6 reports numerical results on the revenue performance of our new algorithm and three other commonly-used heuristics on both the original hotel data and modified data. Section 7 summarizes our research.

### 1.1. Literature Review

The hotel industry and network RM models have been discussed by Goldman et al. (2002) where they showed that a scenario-based LP outperformed deterministic LP by 2-3.5% in a simulated

---

data set. Their scenario-based LP assumes demand for each product follows a discrete distribution with finite support. Liu et al. (2008) also studied stochastic LP and deterministic LP in a simulated data set. Ling et al. (2012) looked at the extended-term stay problem in the hotel RM industry. Our work is most closely related to the network RM and approximate DP literature. Network RM problems are discussed extensively in Talluri and van Ryzin (2004, Chapter 3). Most of the early work on network RM focused on independent demand models (Gallego and van Ryzin 1997, Cooper 2002, Adelman 2007). Even in this simplified case, the resulting optimization problems are difficult to solve in a reasonable amount of time due to the high dimensionality of the state space. Hence, the research has focused on approximation methods, mainly math programming and decomposition approaches.

The most popular math programming approach is deterministic LP (DLP) which was first reported at MIT by Simpson (1989) and Williamson (1992). Talluri and van Ryzin (1999) extended DLP by randomizing realizations of demand, solving the LP associated with each realization and then averaging the bid prices and estimating a gradient from these realizations. They called it RLP (randomized LP) and showed that it improved revenue performance over DLP. Kunnumkal et al. (2012) extended RLP to jointly solve the overbooking and network RM problem.

Bertsimas and Popescu (2003) examined an approximate DP for solving the network RM problem by using the bid prices from the LP relaxation and showed encouraging results over DLP. The earliest work on choice behavior in network RM is the Passenger Origin-Destination Simulator (PODS) study by Bratu (1998) which used the competitive, choice-based environment of PODS to compare a new algorithm he created called probabilistic convergence bid price (ProBP) against traditional LP models. Again, he found revenue improvement in many instances. Zhang and Cooper (2005, 2009) analyze choice among parallel flights in the same origin-destination (OD) market (i.e., different departure times). Their model allows choice between flight times, but not between fare classes (basically assuming effective customer segmentation by fare class restrictions). They present an approximation to DP with effective revenue results. Cooper and Homem-de-Mello (2007) present a Markov decision process (MDP) and solve a simulated six-leg, hub-and-spoke network with effective revenue results. van Ryzin and Vulcano (2008) and Vulcano et al. (2010) used a simulation-based optimization approach on real airline data, under a very general customer choice process and showed a 10% revenue improvement over the airline's current controls. Gallego et al. (2004) proposed an LP approach to analyze network RM for flexible products (i.e., two or more alternatives that an airline could assign a customer to near the end of the booking process [e.g., to a different flight time in the same OD market obviously]) that was subsequently adopted by Liu

---

and van Ryzin (2008) to study network RM with customer choice. They also extend the classic DP decomposition approach to the network setting with customer choice using a multinomial logit (MNL) choice model with disjoint consideration sets.

Recently, the research on network RM with customer choice has been expanding. Tam (2008) tested two different types of DP (Lautenbacher and Stidham (1999), called DPL; and Gallego and van Ryzin (1997), called DP-GVR) in the choice-based, competitive environment of PODS under fully-unrestricted fares in all 500+ OD markets and found that using DPL beat a competitor using EMSR with Q-forecasting by an average of 2.7% in this large network with 84 legs. DP-GVR also outperformed EMSR with Q-forecasting by an average of 2.6% in the same network, but needed both Q-forecasting and fare adjustment (see Fiig et al. (2009) for a description of these two terms) to get such a revenue gain. Miranda Bront et al. (2009) extend the work of Liu and van Ryzin (2008) to allow for overlapping consideration sets in the MNL choice set. Miranda Bront et al. (2009) adopt a DP decomposition approach similar to Liu and van Ryzin (2008). Their paper provides a heuristic solution for the choice-based deterministic LP (CDLP) formulation under a more general choice model, which they showed to have good revenue performance (average revenue improvement of 1.7% over DLP on the 15 cases tested from Williamson’s (1992) large hub-and-spoke network). Zhang and Adelman (2009) adopt a linear functional approximation approach to generate dynamic bid prices that are subsequently used in a DP decomposition. They found a 10.2% average revenue improvement over DLP on the 10 cases they tested in a simulated four-leg, hub-and-spoke network. Kunnumkal and Topaloglu (2008) solve the network RM with customer choice using a Lagrangian relaxation approach to approximate DP. They found an average 0.88% revenue improvement over DLP in 20 cases they tested in a simulated seven-leg, hub-and-spoke network. They extended their work (Kunnumkal and Topaloglu 2010b) to find an approximation for time-dependent and capacity-dependent bid prices that beat leg DP by 0.26% over 27 cases tested in a simulated 11-leg, 2-hub network. Chaneton and Vulcano (2011) solve the network RM problem with customer choice using a stochastic gradient approach. They tested it on real airline data and found an average 2.68% revenue improvement over CDLP in 5 cases.

Even though pricing can be a fundamental control mechanism in network RM, most of the pricing papers in the literature focus on pricing of a single product in isolation, whereas the network RM setting requires pricing multiple itineraries that can substitute for each other. Gallego and van Ryzin (1994), called GvR94 hereafter, analyze the problem of dynamically adjusting the price of a single product, under exponential demand and then characterize the form of the optimal policy. They show that a single-price policy can be asymptotically optimal if both the initial inventory

---

of the product and the length of the selling horizon increase linearly at the same rate. Feng and Gallego (1995) consider the case where the price of a product can be adjusted only once, either from high to low or from low to high. Three different papers [Feng and Gallego (2000), Feng and Xiao (2000) and Zhao and Zheng (2000)] extend the GvR94 analysis by incorporating more complicated demand dynamics and pricing constraints (e.g., more general form of stochastic demand, allowing multiple price reversals, rules that depend on time-to-go and on-hand inventory). Maglaras and Meissner (2006) show that certain pricing problems can be converted into equivalent capacity allocation problems and this allows them to then extend the structural properties for capacity allocation problems to pricing problems.

There is not as much literature when looking at pricing over an entire network. Gallego and van Ryzin (1997) propose a deterministic optimization problem for pricing multiple itineraries that can substitute for each other. They show that the pricing decisions made by this deterministic optimization problem are asymptotically optimal in the exact same way as in GvR94. Kleywegt (2001) developed a joint pricing and overbooking model, where the itinerary requests are deterministic functions of the prices and then solved the model by using Lagrangian duality arguments. Zhang and Cooper (2009) consider the problem of pricing substitutable flights that operate between the same origin-destination pair. They establish upper and lower bounds on the value function and then use these bounds to construct pricing policies, but they do not extend their approach to the more general airline network. Kunnumkal and Topaloglu (2010c) propose a stochastic approximation algorithm for making pricing decisions over an airline network. Erdelyi and Topaloglu (2011) use a variant of the GvR97 approach as a benchmark strategy in their computational experiments (single hub and four or eight spoke cities) and found that their decomposition by leg relaxation technique outperformed deterministic LP (without resolving) by 7.1% on average across 36 test cases. Finally, three overview papers [Bitran and Caldentey (2003), Elmaghraby and Keskinocak (2003), and McGill and van Ryzin (1999)] provide extensive coverage of pricing models in network RM.

As far as literature on price optimization, Li and Huh (2011) consider pricing multiple differentiated products with a Nested Logit model and establish structural results of optimal policies in multi-period dynamic models. Gallego and Wang (2014) look at customers making purchase decisions sequentially under the Nested Logit model-1st they select a nest of products, then they purchase a product within the selected nest (e.g., fare families used by airlines like Air Canada and Air New Zealand). At optimality, they show that adjusted markup is constant for all products within a nest. This reduces the problem dimension to a single variable per nest and they further

show that it then reduces to a single variable optimization of a continuous function. Rayfield et al. (2012) consider variants of a joint pricing problem under the nested logit model and propose several approximation schemes.

## 2. Model Formulation

We consider the dynamic pricing problem in a network with  $m$  resources. The network capacity is denoted by a vector  $c = (c_1, \dots, c_m)$ , where  $c_i$  is the capacity of resource  $i$ ;  $i = 1, \dots, m$ . The resources can be combined to produce  $n$  products. An  $m \times n$  matrix  $A$  is used to represent the resource consumption, where the  $(i, j)$ -th element,  $a_{ij}$ , denotes the quantity of resource  $i$  consumed by one unit of product  $j$ ;  $a_{ij} = 1$  if resource  $i$  is used by product  $j$  and  $a_{ij} = 0$  otherwise. Let  $A_i$  be the  $i$ -th row of  $A$  and  $A^j$  be the  $j$ -th column of  $A$ , respectively. The vector  $A_i$  is also called the product incidence vector for resource  $i$ . Similarly, the vector  $A^j$  is called the resource incidence vector for product  $j$ . To simplify the notation, we use  $j \in A_i$  to indicate that product  $j$  uses resource  $i$  and  $i \in A^j$  to indicate that resource  $i$  is used by product  $j$ . Throughout the paper, we reserve  $i$ ,  $j$ , and  $t$  as the indices for resources, products, and time, respectively.

Customer demand arrives over time. The selling horizon is divided into  $T$  time periods. Time runs forward so that the first time period is period 1, and the last time period is period  $T$ . Period  $T + 1$  is used to represent the end of the selling horizon. In period  $t$ , the probability of one customer arrival is  $\lambda$ , and the probability of no customer arrival is  $1 - \lambda$ . Our formulation can easily accommodate non-stationary arrival probabilities that vary over time by dividing the booking horizon into segments with stationary parameters. The vector  $r$  represents the vector of prices, with  $r_j$  being the price of product  $j$ . Given price  $r$  in time  $t$ , an arriving customer purchases product  $j$  with probability  $P_j(r)$ . We use  $P_0(r)$  to denote the no-purchase probability so that  $\sum_{j=1}^n P_j(r) + P_0(r) = 1$ .

We consider a finite-horizon dynamic programming formulation of the problem. Let  $x$  be the vector of remaining capacity at time  $t$ . Then  $x$  can be used to represent the state of the system. Let  $v_t(x)$  be the maximum expected revenue given state  $x$  at time  $t$ . The Bellman equations can be written as follows:

$$\begin{aligned} \text{(DP)} \quad v_t(x) &= \max_{r_t \in R_t(x)} \left\{ \sum_{j=1}^n \lambda P_j(r_t) [r_{t,j} + v_{t+1}(x - A^j)] + (\lambda P_0(r_t) + 1 - \lambda) v_{t+1}(x) \right\} \\ &= \max_{r_t \in R_t(x)} \left\{ \sum_{j=1}^n \lambda P_j(r_t) [r_{t,j} - \Delta_j v_{t+1}(x)] \right\} + v_{t+1}(x), \end{aligned}$$

where  $\Delta_j v_{t+1}(x) = v_{t+1}(x) - v_{t+1}(x - A^j)$  represents the opportunity cost of selling one unit of product  $j$  in period  $t$ . The boundary conditions are

$$v_{T+1}(x) = 0, \quad \forall x,$$

$$v_t(0) = 0, \quad \forall t.$$

In the above,  $R_t(x) = \times_{j=1}^n R_{t,j}(x)$ , where  $R_{t,j}(x) = \mathfrak{R}_+$  if  $x \geq A^j$  and  $R_{t,j}(x) = \{r_\infty\}$  otherwise. The price  $r_\infty$  is called the null price in the literature (Gallego and van Ryzin 1997). It has the property that  $P_j(r) = 0$  if  $r_j = r_\infty$ . Therefore, when there are not enough resources to satisfy the demand for product  $j$ , the demand is effectively shut off by taking  $r_j = r_\infty$ .

The formulation (DP) can be difficult to analyze mainly for two reasons: the curse of dimensionality and the complexity of the maximization in the Bellman equation. Even if we are able to identify some structural properties, it remains unclear whether they will enable us to solve the problem effectively. Therefore, we focus on heuristic approaches to solve (DP) in the rest of the paper.

### 3. Solution Strategies

#### 3.1. Deterministic Nonlinear Programming Formulation

The use of a deterministic and continuous approximation model has been a popular approach in the RM literature. In the classic network RM setting with fixed prices and independent demand classes, the resulting model is a deterministic linear program, which has been used to construct various heuristic policies to the corresponding dynamic programming models, such as bid-price controls (see Talluri and van Ryzin 1998). Liu and van Ryzin (2008) formulate the deterministic version of the network RM with customer choice as a linear program, which they call the choice-based linear program. Unlike these models, the deterministic approximation of (DP) is a constrained nonlinear programming problem.

In this model, probabilistic and discrete customer arrivals are replaced by continuous and deterministic arrivals with rate  $\lambda$ . Therefore, the model is a deterministic mathematical programming (DMP) model. Given price vector  $r$ , the fraction of customers purchasing product  $j$  is given by  $P_j(r)$ . Let  $d = \lambda T$  be the total customer arrivals over the booking horizon. The deterministic model can be written as

$$(DMP) \quad z_{DMP} = \max_{r \geq 0} \quad d \sum_{j=1}^n r_j P_j(r) \\ \text{s.t.} \quad dAP(r) \leq c. \quad (1)$$

In the formulation above, (1) is a resource constraint where the inequality holds component-wise. The Lagrangian multipliers,  $\pi$ , associated with constraint (1) can be interpreted as the value of an additional unit of each resource. The solution to (DMP) can be used to construct several reasonable heuristics. First, the optimal solution  $r^*$  can be used as the vector of prices. Since  $r^*$  is a constant

vector, which is not time- or inventory-dependent, it results in a static pricing policy where the prices are fixed throughout the selling horizon. Second, the dual values  $\pi$  can be used as bid-prices. Finally, as we will show later, the vector  $\pi$  can be used in a dynamic programming decomposition approach.

Conceptually, (DMP) is the same as the deterministic formulation considered in Gallego and van Ryzin (1997). They show that the solution of the problem is an upper bound on the optimal revenue of (DP). For certain special cases, for example when  $P_j(r)$  is linear, the problem (DMP) is a convex quadratic programming problem, and therefore, is easy to handle. For more general demand functions, the problem is, in general, not convex. However, it can often be transformed into a convex programming problem by a change of variables (Dong et al. 2009).

### 3.2. The Dynamic Programming Decomposition Approach (Zhang and Lu 2013)

The formulation (DP) can be written as a semi-infinite linear program with  $v_t(\cdot)$  as decision variables as follows:

$$\begin{aligned}
 \text{(LP)} \quad & \min_{v_t(\cdot)} v_1(c) \\
 & v_t(x) \geq \sum_{j=1}^n \lambda P_j(r_t) [r_{t,j} + v_{t+1}(x - A^j) - v_{t+1}(x)] + v_{t+1}(x), \quad \forall t, x, r_t \in R_t(x).
 \end{aligned}$$

PROPOSITION 1. *Suppose  $v_t(\cdot)$  solves the optimality equations in (DP) and  $\hat{v}_t(\cdot)$  is a feasible solution to (LP). Then  $\hat{v}_t(x) \geq v_t(x)$  for all  $t, x$ .*

The proof of Proposition 1 follows by induction and is omitted; see Adelman (2007) for details. The formulation (LP) is also difficult to solve because of the huge number of variables and the infinitely many constraints. One way to reduce the number of variables is to use a functional approximation for the value function  $v_t(\cdot)$ ; see Adelman (2007).

Zhang and Lu (2013) introduce a dynamic programming decomposition approach to solve (DP) based on the dual variables  $\pi$  in (DMP). For each  $i, t$ , and  $x$ ,  $v_t(x)$  can be approximated by

$$v_t(x) \approx v_{t,i}(x_i) + \sum_{k \neq i} x_k \pi_k. \quad (2)$$

Therefore, the value  $v_t(x)$  is approximated by the sum of a nonlinear term for resource  $i$  and linear terms for all other resources. Note  $v_{t,i}(x_i)$  can be interpreted as the approximate value of  $x_i$  units of resource  $i$ , and  $x_k \pi_k$  can be interpreted as the value of resource  $k$ . Using (2) in (DP) and simplifying, we obtain

$$\text{(DP}_i\text{)} \quad v_{t,i}(x_i) = \max_{r_t \in R_t(x_i, c_{-i})} \left\{ \sum_{j=1}^n \lambda P_j(r_t) \left[ r_{t,j} - \sum_{k \neq i} a_{kj} \pi_k + v_{t+1,i}(x_i - a_{ij}) - v_{t+1,i}(x_i) \right] \right\} + v_{t+1,i}(x_i)$$

$$= \max_{r_t \in R_t(x_i, c_{-i})} \left\{ \sum_{j=1}^n \lambda P_j(r_t) \left[ r_{t,j} - \sum_{k \neq i} a_{kj} \pi_k - \Delta_j v_{t+1,i}(x_i) \right] \right\} + v_{t+1,i}(x_i).$$

The boundary conditions are

$$\begin{aligned} v_{T+1,i}(x) &= 0, & \forall x, \\ v_{t,i}(0) &= 0, & \forall t. \end{aligned}$$

In the above,  $(x_i, c_{-i})$  is an  $m$ -vector whose  $i$ -th component is  $x_i$  and  $k$ -th component is  $c_k$  for  $k \neq i$ . The set of  $m$  one-dimensional dynamic programs can be solved to obtain the values of  $v_{t,i}(x_i)$  for each  $i$ .

Zhang and Lu (2013) show that the approximation scheme (2) yields an upper bound, which is tighter than the bound from the deterministic formulation.

**PROPOSITION 2 (Zhang and Lu (2013)).** *For each  $i$ , let  $v_{t,i}^*(\cdot)$  be an optimal solution and a feasible solution to  $(\mathbf{LP}_i)$ . Then, for each  $k = 1, \dots, m$ , we have*

$$z_{DMP} \geq v_{1,k}^*(c_k) + \sum_{k \neq i} c_k \pi_k \geq \min_i \left\{ v_{1,i}^*(c_i) + \sum_{k \neq i} c_k \pi_k \right\} \geq v_1(c).$$

### 3.3. A New Dynamic Programming Decomposition Approach

In this section, we generalize the nonlinear non-separable approximation developed in Zhang (2011) to the dynamic pricing setup. This approach leads to an improved dynamic programming decomposition. Given a vector  $\pi^*$  of resource dual prices and a collection of single-dimensional value functions  $\{\hat{v}_{t,i}(\cdot)\}_{\forall t,i}$ , we consider the functional approximation

$$v_t(x) \approx \min_i \left\{ \hat{v}_{t,i}(x_i) + \sum_{k \neq i} \pi_k^* x_k \right\}, \quad \forall t, x. \quad (3)$$

In the above,  $\hat{v}_{t,i}(x_i)$  is a nonlinear term representing the value of  $x_i$  seats on resource  $i$ . Observe that the approximation (3) is nonlinear and non-separable in  $x$  due to the minimization. The approximation (3) is able to better capture the network effect because after the  $\hat{v}_{t,i}(\cdot)$ 's are determined for each resource  $i$ , the value  $v_t(x)$  is approximated by a single minimum across the resources.

Plugging (3) into (LP), we have

$$\begin{aligned} (\text{NLP}) \quad z_{NLP} &= \min_{\hat{v}_{t,i}(\cdot) \forall t,i} \min_i \left\{ \hat{v}_{1,i}(c_i) + \sum_{k \neq i} \pi_k^* c_k \right\} \\ &\min_i \left\{ \hat{v}_{t,i}(x_i) + \sum_{k \neq i} \pi_k^* x_k \right\} \geq \sum_{j=1}^n \lambda P_j(r_t) \left( r_{t,j} + \min_l \left\{ \hat{v}_{t+1,l}(x_l - a_{lj}) + \sum_{k \neq l} \pi_k^* (x_k - a_{kj}) \right\} \right) \\ &+ (\lambda P_0(r_t) + 1 - \lambda) \min_l \left\{ \hat{v}_{t+1,l}(x_l) + \sum_{k \neq l} \pi_k^* x_k \right\}, \quad \forall t, x, r_t \in R_t(x). \end{aligned} \quad (4)$$

First, observe that the minimization on the left-hand side of (4) can be removed by writing each constraint as  $m$  equivalent constraints

$$\begin{aligned} \hat{v}_{t,i}(x_i) + \sum_{k \neq i} \pi_k^* x_k \geq & \sum_{j=1}^n \lambda P_j(r_t) \left( r_{t,j} + \min_l \left\{ \hat{v}_{t+1,l}(x_l - a_{lj}) + \sum_{k \neq l} \pi_k^* (x_k - a_{kj}) \right\} \right) \\ & + (\lambda P_0(r_t) + 1 - \lambda) \min_l \left\{ \hat{v}_{t+1,l}(x_l) + \sum_{k \neq l} \pi_k^* x_k \right\}, \quad \forall i, t, x, r_t \in R_t(x). \end{aligned} \quad (5)$$

The constraint (5) is still difficult to handle. In the following, we consider a restriction of (5) that renders the resulting problem efficiently solvable via a simultaneous dynamic programming approach.

By moving the second term on the left-hand side to the right-hand side, the constraint (5) can be written as

$$\begin{aligned} \hat{v}_{t,i}(x_i) \geq & \sum_{j=1}^n \lambda P_j(r_t) \left( r_{t,j} + \min_l \left\{ \hat{v}_{t+1,l}(x_l - a_{lj}) + \sum_{k \neq l} \pi_k^* (x_k - a_{kj}) - \sum_{k \neq i} \pi_k^* x_k \right\} \right) \\ & + (\lambda P_0(r_t) + 1 - \lambda) \min_l \left\{ \hat{v}_{t+1,l}(x_l) + \sum_{k \neq l} \pi_k^* x_k - \sum_{k \neq i} \pi_k^* x_k \right\}, \quad \forall i, t, x, r_t \in R_t(x). \end{aligned}$$

Simplifying the above leads to

$$\begin{aligned} \hat{v}_{t,i}(x_i) \geq & \sum_{j=1}^n \lambda P_j(r_t) \left( r_{t,j} + \min_l \left\{ \hat{v}_{t+1,l}(x_l - a_{lj}) - \pi_l^* x_l - \sum_{k \neq l} \pi_k^* a_{kj} + \pi_i^* x_i \right\} \right) \\ & + (\lambda P_0(r_t) + 1 - \lambda) \min_l \left\{ \hat{v}_{t+1,l}(x_l) - \pi_l^* x_l + \pi_i^* x_i \right\}, \quad \forall i, t, x, r_t \subseteq R_t(x). \end{aligned}$$

Breaking up the minimization terms on the right-hand side leads to

$$\begin{aligned} \hat{v}_{t,i}(x_i) \geq & \sum_{j=1}^n \lambda P_j(S) \left( r_{t,j} + \min \left\{ \hat{v}_{t+1,i}(x_i - a_{ij}) - \sum_{k \neq i} \pi_k^* a_{kj}, \right. \right. \\ & \left. \left. \min_{l \neq i} \left\{ \hat{v}_{t+1,l}(x_l - a_{lj}) - (x_l - a_{lj}) \pi_l^* - \sum_k a_{kj} \pi_k^* + \pi_i^* x_i \right\} \right\} \right) \\ & + (\lambda P_0(r_t) + 1 - \lambda) \min \left\{ \hat{v}_{t+1,i}(x_i), \min_{l \neq i} \left\{ \hat{v}_{t+1,l}(x_l) - \pi_l^* x_l + \pi_i^* x_i \right\} \right\}, \quad \forall i, t, x, r_t \in R_t(x). \end{aligned}$$

Next, we restrict the constraint such that, for each fixed  $i$ , the constraint only involves  $x_i$ , which can be achieved by taking the maximum over  $x_k$  for all  $k \neq i$  for each fixed  $i$ . With a little more algebra, this leads to

$$\begin{aligned} \hat{v}_{t,i}(x_i) \geq & \sum_{j=1}^n \lambda P_j(r_t) \left( r_{t,j} + \min \left\{ \hat{v}_{t+1,i}(x_i - a_{ij}) - \sum_{k \neq i} \pi_k^* a_{kj}, \right. \right. \\ & \left. \left. \min_{l \neq i} \left\{ \max_{0 \leq y_l \leq c_l - a_{lj}} [\hat{v}_{t+1,l}(y_l) - y_l \pi_l^*] - \sum_k a_{kj} \pi_k^* + \pi_i^* x_i \right\} \right\} \right) \\ & + (\lambda P_0(r_t) + 1 - \lambda) \min \left\{ \hat{v}_{t+1,i}(x_i), \min_{l \neq i} \left\{ \max_{0 \leq y_l \leq c_l} [\hat{v}_{t+1,l}(y_l) - \pi_l^* y_l] + \pi_i^* x_i \right\} \right\}, \end{aligned}$$

$$\forall i, t, x, r_t \in R_t(x).$$

Recognizing that  $R(x) \subseteq R(x_i, c_{-i})$  with  $c_{-i}$  being the vector  $c$  without the  $i$ -th component, the above constraint can be further restricted by taking  $r_t \in R_t(x_i, c_{-i})$  instead of  $r_t \in R_t(x)$ . Note that this is a constraint restriction because for each fixed  $i, t$ , and  $x$ , replacing  $R_t(x)$  with  $R_t(x_i, c_{-i})$  adds constraints for all  $r_t \in R_t(x_i, c_{-i}) \setminus R_t(x)$ . After this step, we can remove redundant constraints by replacing  $x$  with  $x_i$  for each  $i$ , since the constraints for each  $i$  do not involve other components of  $x$  except  $x_i$ . For convenience of reference, we rewrite (NLP) with the restricted constraint as

$$\begin{aligned} (\widehat{\text{NLP}}) \quad z_{\widehat{\text{NLP}}} &= \min_{\hat{v}_{t,i}(\cdot) \forall t,i} \min_i \left\{ \hat{v}_{1,i}(c_i) + \sum_{k \neq i} \pi_k^* c_k \right\} \\ \hat{v}_{t,i}(x_i) &\geq \sum_{j=1}^n \lambda P_j(r_t) \left( r_{t,j} + \min \left\{ \hat{v}_{t+1,i}(x_i - a_{ij}) - \sum_{k \neq i} \pi_k^* a_{kj}, \right. \right. \\ &\quad \left. \left. \min_{l \neq i} \left\{ \max_{0 \leq y_l \leq c_l - a_{lj}} [\hat{v}_{t+1,l}(y_l) - y_l \pi_l^*] - \sum_k a_{kj} \pi_k^* + \pi_i^* x_i \right\} \right\} \right) \\ &\quad + (\lambda P_0(r_t) + 1 - \lambda) \min \left\{ \hat{v}_{t+1,i}(x_i), \min_{l \neq i} \left\{ \max_{0 \leq y_l \leq c_l} [\hat{v}_{t+1,l}(y_l) - \pi_l^* y_l] + \pi_i^* x_i \right\} \right\}, \\ &\quad \forall i, t, x_i, r_t \subseteq R_t(x_i, c_{-i}). \end{aligned} \tag{6}$$

From the construction of the restricted constraints, all feasible solutions to  $(\widehat{\text{NLP}})$  are also feasible for (NLP). The following proposition is a direct consequence of this observation.

PROPOSITION 3.  $z_{\widehat{\text{NLP}}} \geq z_{\text{NLP}}$ .

The formulation  $(\widehat{\text{NLP}})$  is still a nonlinear optimization problem with a large number of nonlinear constraints. Therefore, a brute-force solution of the problem is very computationally intensive, if possible at all. Next, we introduce a simultaneous dynamic programming approach to solve  $(\widehat{\text{NLP}})$ . We will show that the approach leads to an efficient solution algorithm for the problem  $(\widehat{\text{NLP}})$ . Furthermore, the approach also leads to a simple inductive proof of the relationships among the different bounds. We first show the following proposition.

PROPOSITION 4. *For each feasible solution  $\{\hat{v}_{t,i}(\cdot)\}_{\forall t,i}$  of  $(\widehat{\text{NLP}})$ , there exists a feasible solution with equal or smaller objective value such that equality in (6) holds for some  $\tilde{r}_{t,i}(x_i) \in R(x_i, c_{-i})$  for each  $t, i, x_i$ .*

Proof. Let  $\{\hat{v}_{t,i}\}_{\forall t,i}$  be a feasible solution to  $(\widehat{\text{NLP}})$ . Suppose  $\hat{v}_{t,i}(x_i)$  is strictly greater than the right-hand side of (6) for all  $S \subseteq N(x_i, c_{-i})$  for fixed  $t, i, x_i$ . Then, the solution  $\{\hat{v}_{t,i}\}_{\forall t,i}$  can be modified by replacing  $\hat{v}_{t,i}(x_i)$  with

$$v_{t,i}^\dagger(x_i) = \max_{r_t \subseteq N(x_i, c_{-i})} \sum_{j=1}^n \lambda P_j(r_t) \left( r_{t,j} + \min \left\{ \hat{v}_{t+1,i}(x_i - a_{ij}) - \sum_{k \neq i} \pi_k^* a_{kj}, \right. \right.$$

$$\min_{l \neq i} \left\{ \max_{0 \leq y_l \leq c_l - a_{lj}} [\hat{v}_{t+1,l}(y_l) - y_l \pi_l^*] - \sum_k a_{kj} \pi_k^* + \pi_i^* x_i \right\} \Bigg\} \\ + (\lambda P_0(r_t) + 1 - \lambda) \min \left\{ \hat{v}_{t+1,i}(x_i), \min_{l \neq i} \left\{ \max_{0 \leq y_l \leq c_l} [\hat{v}_{t+1,l}(y_l) - \pi_l^* y_l] + \pi_i^* x_i \right\} \right\}.$$

It is easy to check that the new solution is still feasible. Repeating this procedure for all values in the solution such that strict inequality holds in (6) yields the result.  $\blacksquare$

An immediate corollary for Proposition 4 is the following.

**COROLLARY 1.** *There exists an optimal solution  $\{\hat{v}_{t,i}(\cdot)\}_{\forall t,i}$  to  $(\widehat{NLP})$  such that equality holds for some  $\tilde{r}_{t,i}(x_i) \in R_t(x_i, c_{-i})$  for each fixed  $t, i, x_i$ .*

Observe that, by construction, each constraint in  $(\widehat{NLP})$  only involves  $x_i$  for one resource  $i$ , but not for all other resources. Define a set of value functions  $\{\tilde{v}_{t,i}(\cdot)\}$  for all  $t$  and  $i$  as follows:

$$(\widehat{DP}) \quad \tilde{v}_{t,i}(x_i) = \max_{r_t \in R_t(x_i, c_{-i})} \sum_{j=1}^n \lambda P_j(r_t) \left( r_{t,j} + \min \left\{ \tilde{v}_{t+1,i}(x_i - a_{ij}) - \sum_{k \neq i} \pi_k^* a_{kj}, \right. \right. \\ \left. \left. \min_{l \neq i} \left\{ \max_{0 \leq y_l \leq c_l - a_{lj}} [\tilde{v}_{t+1,l}(y_l) - y_l \pi_l^*] - \sum_k a_{kj} \pi_k^* + \pi_i^* x_i \right\} \right\} \right) \\ + (\lambda P_0(r_t) + 1 - \lambda) \min \left\{ \tilde{v}_{t+1,i}(x_i), \min_{l \neq i} \left\{ \max_{0 \leq y_l \leq c_l} [\tilde{v}_{t+1,l}(y_l) - \pi_l^* y_l] + \pi_i^* x_i \right\} \right\}, \\ \forall i, t, x_i$$

with boundary conditions  $\tilde{v}_{T+1,i}(x_i) = 0$  for all  $i, x_i$ .

Next, we establish the equivalence between  $(\widehat{NLP})$  and  $(\widehat{DP})$ . We have the following result:

**PROPOSITION 5.** *There exists an optimal solution  $\{\hat{v}_{t,i}^*(\cdot)\}_{\forall t,i}$  to  $(\widehat{NLP})$  such that  $\tilde{v}_{t,i}(x_i) = \hat{v}_{t,i}^*(x_i)$  for all  $t, i, x$ , where  $\tilde{v}_{t,i}(x_i)$  is defined in  $(\widehat{DP})$ .*

*Proof.* From Corollary 1, it is without loss of optimality to restrict our attention in  $(\widehat{NLP})$  to optimal solutions such that equalities hold in the constraint (6). Let  $\{\hat{v}_{t,i}^*(x_i)\}_{\forall t,i}$  denote such an optimal solution to  $(\widehat{NLP})$ . Then,

$$\hat{v}_{t,i}^*(x_i) = \max_{r_t \in R_t(x_i, c_{-i})} \sum_{j=1}^n \lambda P_j(r_t) \left( r_{t,j} + \min \left\{ \hat{v}_{t+1,i}^*(x_i - a_{ij}) - \sum_{k \neq i} \pi_k^* a_{kj}, \right. \right. \\ \left. \left. \min_{l \neq i} \left\{ \max_{0 \leq y_l \leq c_l - a_{lj}} [\hat{v}_{t+1,l}^*(y_l) - y_l \pi_l^*] - \sum_k a_{kj} \pi_k^* + \pi_i^* x_i \right\} \right\} \right) \\ + (\lambda P_0(r_t) + 1 - \lambda) \min \left\{ \hat{v}_{t+1,i}^*(x_i), \min_{l \neq i} \left\{ \max_{0 \leq y_l \leq c_l} [\hat{v}_{t+1,l}^*(y_l) - \pi_l^* y_l] + \pi_i^* x_i \right\} \right\} \\ \forall i, t, x_i. \tag{7}$$

Comparing (7) with  $(\widehat{DP})$  yields the result.  $\blacksquare$

Because of the equivalence of  $\tilde{v}_{t,i}(x_i)$  in  $(\widehat{DP})$  and  $\hat{v}_{t,i}^*(x_i)$ , we will only use  $\hat{v}_{t,i}^*(x_i)$  in the remainder of the paper. Proposition 5 shows that it suffices to solve  $(\widehat{DP})$  instead of  $(\widehat{NLP})$ . The formulation  $(\widehat{DP})$  involves  $m$  single-resource dynamic programs that need to be solved simultaneously because the optimality equation for  $\hat{v}_{t,i}^*(x_i)$  involves  $\hat{v}_{t+1,k}^*(x_k)$  for all  $x$  and  $k = 1, \dots, m$ .

We show that the bounds from  $(\widehat{NLP})$  are tighter than the decomposition bound in Proposition 2 originally shown in Zhang and Lu (2013).

PROPOSITION 6. *The following results hold:*

- (i)  $\hat{v}_{t,i}^*(x_i) \leq v_{t,i}^*(x_i), \forall i, x_i;$
- (ii)  $v_1(c) \leq z_{NLP} \leq z_{\widehat{NLP}} = \min_i \left\{ \hat{v}_{1,i}^*(c_i) + \sum_{k \neq i} \pi_k^* c_k \right\} \leq \min_i \left\{ v_{1,i}^*(c_i) + \sum_{k \neq i} \pi_k^* c_k \right\} \leq z_{DMP}.$

## 4. Case Study at a Major Hotel

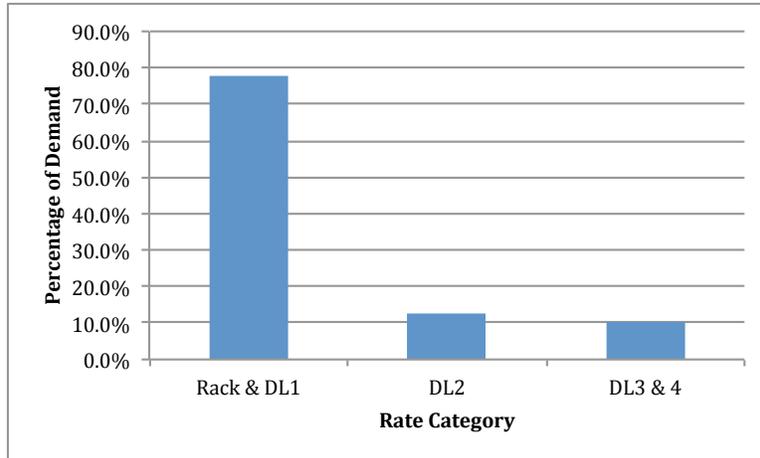
We test the approach proposed in this paper on real hotel data (i.e., a major hotel property with capacity of 2000+ rooms, five different rate categories, 16 snapshot points or booking periods, varying lengths of stay [LOS] between one night and up to 12 consecutive nights, over a 30 day actual arrival horizon of stay nights from mid-July to mid-August).

### 4.1. Problem Description and Data

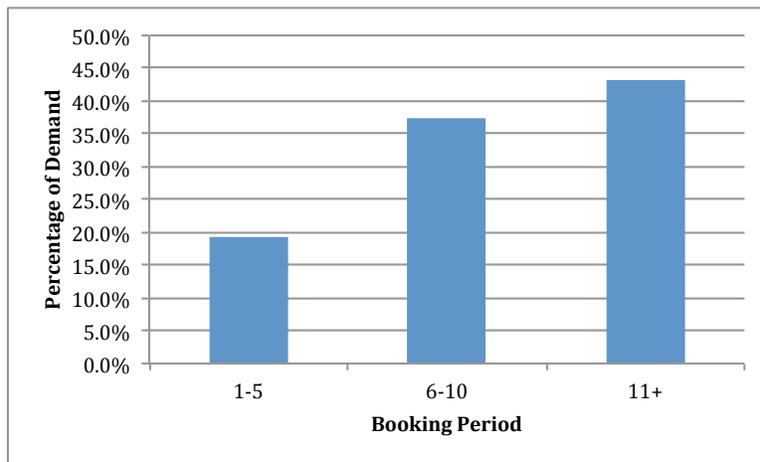
Figure 1 shows how the demand is distributed across the rate category groups (Rack and discount level 1 [DL1], discount level 2 [DL2], discount level 3 and 4 [DL3/4]).<sup>3</sup> As can be seen, approximately 77% of total demand is obtained at the highest two rates, with about another 13% in the next discount level, DL2. Also, the bottom two rate categories, DL3 and DL4 account for about 10% of total demand. The discounts range from 0 to 50% off the rack rate. The average rack rate for this property and room type is \$150 per night.

Figure 2 shows the mix of how reservations for demand arrive across the 15+ different snapshot or booking periods before the actual stay night. This hotel property starts accepting reservations up to two full years in advance of the stay night and splits up the remaining days before the actual stay night into 15+ booking periods. Period 1 is the one farthest in advance, while the last period represents the time period just before the actual arrival. As can be seen, approximately 19% of total reservations for demand have typically arrived in the first five periods. Periods 6-10 have about 37% of reservation demand. Finally, the last grouping has the highest percentage with approximately 43% arriving. Of course, to implement this with a DP approach, the 15+ booking periods need to be further subdivided into tens of thousands of reservation arrival periods, adding to the enormous complexity of the network problem and its solution.

<sup>3</sup> Due to a confidentiality agreement, we have to suppress some details when reporting the data.



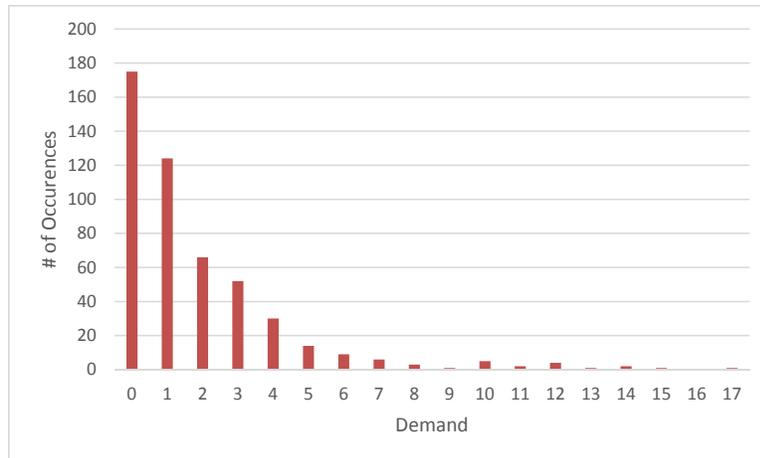
**Figure 1** Percentage of demand requests at different rate categories



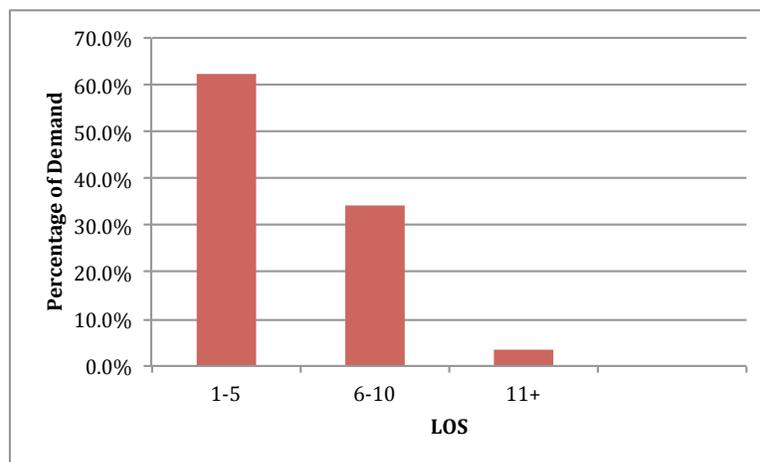
**Figure 2** Percentage of demand arriving by booking period

The arrivals data is assumed to follow a Poisson process. This assumption can be verified through a goodness-of-fit test. To give an example, we selected the observed values for a typical LOS (i.e., 7 night stay) and typical rate category (i.e., Rack rate). The observed demand values (for all the arrival days and booking periods) are shown in Figure 3. Performing a Chi-square test on these actuals vs. 496 randomly generated Poisson values using the same mean (1.863), generates a chi-squared value of 226.11 with 495 degrees of freedom. The probability therefore that this arrivals data comes from a Poisson process is 0.9999, or in other words, it has a very good fit with the Poisson distribution.

Figure 4 shows the mix of demand across the different potential lengths of stay (LOS). As can be seen, a little over 60% of customers stay for one to five nights (the biggest grouping). Six to 10 night stays generate 34% of total demand, while stays of 11 or more nights fall off dramatically and only equal 3% of total demand. Unlike the typical airline example, where you generally have a one



**Figure 3** Observed rack rate demand for LOS=7 over all arrival days and booking periods.



**Figure 4** Percentage of demand by length-of-stay (LOS)

leg flight (non-stop) or a two-leg (connecting) flight and the most you might see would be three legs, this hotel example clearly has plenty of demand for five, six or seven "connecting" resources (stay nights). Important ingredients of a DP model include resources, products, and a demand forecast. Specific to the hotel industry, resources correspond to hotel rooms on different stay nights. A product is a one- or multiday stay at a particular room rate category.

The data exhibits non-stationarity in two different ways. First the demand that arrives on the different day-of-week stay nights (e.g., Friday vs. Monday) is of different magnitude. Second, as part of the build-up curve, different booking periods show stronger or weaker demand rates as you approach the stay night as reflected in Figure 2.

The major hotel has used the standard 'split data' approach to check the validity of the estimated

parameters, that is, roughly half the data was used to estimate the different parameters, while the remaining 50% was used as a hold-out sample to test the validity of the initial estimates.

## 4.2. Problem Formulation

It is well-known in the revenue management literature that a hotel revenue management problem can be mapped to a network revenue management formulation; see, for example, Gallego and van Ryzin (1997). Nevertheless, there are some important differences between hotel revenue management problems and airline revenue management problems that motivated the original formulation. In particular, the beginning and end of a sales horizon are well-defined for airline problems, while it is not as clear in the hotel problem because different room-nights are linked by customers who stay multiple nights. For this reason, it is reasonable to view a hotel revenue management problem as an infinite horizon problem. However, a popular approach in the hotel industry is to formulate and solve the RM problem as a rolling horizon, finite-horizon problem, while the end-of-horizon is determined by a cut-off date. Typically, the cut-off date is a few hundred days in the future. This calls for special processing for the end-of-horizon.

Due to data limitations, our numerical study considers a cut-off date of 30 days in the future. Therefore, the scale of the problem we consider is smaller than problems solved in practice. We note, however, the solution time of decomposition methods introduced in this paper is roughly linear in the number of resources. Hence, our numerical study gives a good sense of the solution time.

In order to properly handle end-of-horizon demand, we alter the length of stay to maintain the same demand load within the booking horizon. All demand in the last day is assigned a length of stay of 1, because any demand with length of stay greater than 1 will use resources not considered in the problem formulation. Similarly, all demand in the second to last day with a length of stay greater than 2 is assigned a length of stay of 2. This is done for all days where a demand request may use resources outside of the booking horizon we consider.

Even though we develop theoretical results for a rather general demand model, the demand model used at the hotel is fairly specific, and corresponds to what is known in the literature as *priceable demand* (Boyd and Kalllesen 2004). In the model developed in Section 2, the choice probability of product  $j$  is given by  $P_j(r)$ , where  $r$  is the vector of prices for all products. This choice probability implies that demand for each product depends on the prices of all products. Our numerical implementation considers two restrictions: (i) the demand for each product only depends on the price of that product, and (ii) there are a discrete number of price levels. Let  $k = 1, \dots, K$  denote the price level. It is assumed that the price for each night-stay is  $f_k$  when price level  $k$

is used and an  $l$ -night stay costs  $lf_k$ . The price level is ordered such that  $f_1 > f_2 > \dots > f_k$ . The booking horizon is divided into  $H$  decision control periods (DCPs). The demand profile is assumed to be stationary within each DCP but can be different across DCPs.

Let  $d_{hilk}$  denote the *incremental* demand in DCP  $h$  at price level  $k$  for customers whose stay starts on day  $i$  and ends on day  $l$ . When  $k = 1$ ,  $d_{hil1}$  simply denotes the demand in DCP  $h$  at price level 1 for customers whose stay starts on day  $i$  and ends on day  $l$ . For  $k > 1$ ,  $d_{hilk}$  refers to the incremental demand by reducing price from level  $k - 1$  to level  $k$ . Therefore, the total demand at price level  $k$  is  $\sum_{k'=1}^k d_{hilk'}$ .

We use the following linear programming formulation:

$$(LP) \quad \max_x \sum_h \sum_i \sum_l \sum_k (l-i+1) f_k w_{hilk} \left( \sum_{k'=1}^k d_{hilk'} \right) \quad (8)$$

$$\sum_h \sum_{i'=i-11}^i \sum_{l=i-i'+1}^{i+12} \sum_k w_{hilk} \left( \sum_{k'=1}^k d_{hilk'} \right) \leq c_i, \quad \forall i, \quad (8)$$

$$\sum_k w_{hilk'} \leq 1, \quad \forall h, i, l, \quad (9)$$

$$w \geq 0.$$

In the above, the decision variable  $w_{hilk}$  is the fraction of time that price level  $k$  is used in DCP  $h$  for customers who stay from day  $i$  to day  $l$ .

The formulation (LP) is the same as the one used in Erdelyi and Topaloglu (2011), where they also show that the objective value of (LP) gives an upper bound on the total expected revenue.

## 5. Control Policies

In this section, we introduce heuristic policies from  $(\widehat{DP})$ . For comparison purposes, we also consider policies from (LP) and the classical dynamic programming decomposition (Zhang and Lu 2013).

### 5.1. DLP

Let  $\pi^*$  be the vector of dual values corresponding to the resource constraints in (LP). In our numerical study, we call the policy that approximates  $\Delta_j v_i(x)$  with  $\sum_i a_{ij} \pi_i^*$  for each  $j$  for all  $t$  and  $x \geq A^j$  DLP. Note that product  $j$  will not be offered when  $x_i < a_{ij}$  for some  $i$ . Unlike bid-price policies for independent demand models (Talluri and van Ryzin 1998), it is possible that a product  $j$  with  $r_{t,j} > \sum_i a_{ij} \pi_i^*$  may not be offered. DLP has the lowest computational cost in the policies we test in this paper as, unlike the decomposition-based approaches, it does not involve the computation of dynamic programming value functions. One way to improve the performance of policies from static approximations, such as DLP, is to resolve the static model taking into account changes in the capacity and remaining time. In our numerical study, we also consider a version of

DLP, where the bid-prices are updated 16 times, by resolving at the beginning of each booking period.

## 5.2. DCOMP

This policy implements the dynamic programming decomposition introduced in Section 3.2. After the collection of value functions  $\{v_{t,i}(\cdot)\}_{\forall t,i}$  is computed, the value function  $v_t(x)$  can then be approximated by

$$v_t(x) \approx \sum_{i=1}^m v_{t,i}(x_i). \quad (10)$$

By using (10), we have

$$\Delta_j v_t(x) = v_t(x) - v_t(x - A^j) \equiv \sum_{i=1}^m \Delta_j v_{t,i}(x_i). \quad (11)$$

An approximate policy to (DP) is given by

$$r_t^*(x) = \arg \max_{r_t \in R_t(x)} \left\{ \sum_{j=1}^n \lambda P_j(r_t) \left[ r_{t,j} - \sum_{i=1}^m \Delta_j v_{t+1,i}(x_i) \right] \right\}. \quad (12)$$

## 5.3. DCOMP-AVG

We also consider a variant of **DCOMP** proposed in Erdelyi and Topaloglu (2011). Instead of using (11) to approximate the marginal value, they consider the approximation

$$\Delta_j v_t(x) = \frac{1}{\sum_i a_{ij}} \sum_i a_{ij} \left[ v_{t,i}(x) - v_{t,i}(x - a_{ij}) + \sum_{i' \neq i}^m a_{i'j} \pi_i^* \right].$$

Then (12) is used to generate a control policy. Erdelyi and Topaloglu (2011) show promising performance on a set of randomly generated problem instances.

## 5.4. DCOMP1

In this subsection, we introduce a heuristic policy from the solution of  $(\widehat{DP})$ . Recall that the solution is denoted by  $\{\hat{v}_{t,i}^*(\cdot)\}_{\forall i,t}$ . Consider the separable approximation  $v_t(x) \approx \sum_i \hat{v}_{t,i}^*(x_i), \forall t, x$ . Then, the opportunity cost of selling product  $j$  in period  $t$  for  $x \geq A^j$  is given by

$$\Delta_j v_{t+1}(x) = v_{t+1}(x) - v_{t+1}(x - A^j) \approx \sum_i \hat{v}_{t+1,i}^*(x_i) - \sum_i \hat{v}_{t+1,i}^*(x_i - a_{ij}) \equiv \tilde{\Delta}_j v_{t+1}(x).$$

By replacing  $\Delta_j v_{t+1}(x)$  in (DP) with  $\tilde{\Delta}_j v_{t+1}(x)$  for all  $j$ , the prices in period  $t$  are given by

$$r_t^\dagger(x) = \arg \max_{r_t \in R_t(x)} \left\{ \sum_{j=1}^n \lambda P_j(r_t) \left[ r_{t,j} - \sum_{i=1}^m \tilde{\Delta}_j v_{t+1,i}(x_i) \right] \right\}.$$

A heuristic policy that offers prices  $r_t^\dagger(x)$  in period  $t$  and state  $x$  is called **DCOMP1**, in our numerical study.

## 6. Numerical Results

In this section, we report results from a numerical study that investigates the revenue and computational performance of the policy DCOMP1 introduced in Section 5.4. The performance of DCOMP1 is compared to DLP (Section 5.1), DCOMP (Section 5.2), and DCOMP-AVG (Section 5.3). We also implemented a version of DLP that re-solves 16 times before the stay night, at the beginning of each DCP, which we call DLP16. Because DLP policy without resolving does not perform well in our numerical study, we dropped the results in the remainder of the paper, other than Table 3.

### 6.1. Description of Cases from Original Hotel Data and Bound Performance

**6.1.1. Description of Cases** To study the revenue and upper bound performance, we take the base case described in Section 4 and create nine additional cases that vary methodically in terms of the capacity scale factor (csf). That is, we leave the demand as is and we alter the capacity of the hotel. In the base case (csf = 1.0), the hotel has 200 rooms available each night over the 30 night horizon. In the most extreme case (#1), we downsize the hotel to 80 rooms or a csf = 0.4. This obviously creates the situation where we have the highest potential demand-to-capacity ratio. On the other extreme (case #10), we upsize the hotel to 260 rooms or csf = 1.3. See Table 1 for details on all ten cases. The demand-to-capacity column (last one) shows the ratio of potential demand relative to the available capacity in each case, which is given by

$$\rho = \frac{\sum_i \sum_h \sum_{i'=i-11}^i \sum_{l=i-i'+1}^{12} \left( \sum_{k=1}^5 d_{hilk} \right)}{\sum_i c_i}.$$

**6.1.2. Bounds** Table 2 reports the upper bounds for cases 1 to 10. We report three bounds: the DLP objective value, the decomposition bound from DCOMP, and the decomposition bound from DCOMP1. The bound improvement columns show the percentage difference between the DCOMP1 bound and the other two bounds. The three bounds are very close to each other and are consistent with their theoretical relationships, with the DLP bound the largest and DCOMP1 bound the lowest. The fact that the three bounds are close is simply the consequence of the magnitude of the problem instances; as we know, all three bounds are asymptotically optimal as the problem size scales up.

### 6.2. Policy Performance of Algorithms on Original Data

The revenue performance of the four different policies is evaluated through simulation. We randomly generated 1,000 demand streams for each case where each demand stream contains a random variable for each time period. The same random demand stream is used for all simulated policies in each case.

**6.2.1. Revenue Results from Ten Cases** The four columns of Table 3 measure the percentage revenue gains of DCOMP1 against the four benchmark policies—DLP, DLP16, DCOMP and DCOMP-AVG. The results show that DCOMP1 outperforms DLP by as much 3.25% and an average of 1.61% across the ten cases. They also show that DCOMP1 outperforms a frequently-solved DLP algorithm (DLP16) and DCOMP by as much as 0.05% and 0.15% respectively. Finally it outperforms DCOMP-AVG by as much as 0.48% and an average of 0.19% across the ten cases. Given that case #7 is the base case (where demand is about equal to capacity), we would consider cases #5 to 9 to be the sweet spot of opportunity for RM algorithms. Although it may not seem like much, depending on the problem instance, this truly can be a significant improvement, especially given that DP decomposition is the state-of-the-art in the industry in terms of implemented algorithms. For every \$1 billion revenue generated by a hotel company, an improvement of 0.19% would represent an increase in revenue of \$1.9 million, most of which would drop to the bottom line (given the low variable cost of having an extra room occupied). In general we see that for very low demand-to-capacity ratios, the percentage revenue improvement approaches zero (e.g., case 10).

Table 4 reports the empirical results for average occupancy (i.e., what percentage of the rooms are occupied on any given stay night), which is an important industry metric. We note that DCOMP1 has slightly lower average occupancies than DCOMP (0.1-0.3%) in the first 5 cases; then it has slightly higher occupancies (0.3-0.8%) in the next 4 cases. Compared to DLP16, DCOMP1 generally has lower average occupancies (0.2-0.7%) in the majority of the cases. Compared to DCOMP-AVG, DCOMP1 generally has higher average occupancies (0.1-0.8%) in the majority of the cases. Finally, Table 5 reports both average daily room rate (ADR) per night (again, recall that rack rate = \$150) and the average length-of-stay (LOS) of a guest. We see that DCOMP1 has a slightly higher (average of \$0.11) ADR than DCOMP for the 1st 5 cases, then slightly lower (average of \$0.66) for the next 4 cases. The results also show that DCOMP1 has a slightly higher (average of \$0.25) ADR than DLP16 in 7 out of 10 cases and slightly lower (average of \$0.23) ADR than DCOMP-AVG in 8 out of 10 cases.

As we know, in order to maximize revenue it takes a combination of Tables 4 and 5. In all 10 cases where DCOMP1 beats DCOMP-AVG, it does so with higher occupancies and lower ADRs. In cases when DCOMP1 beats DCOMP (7 cases), it generally does so with higher occupancies and lower ADRs as well. Finally, when DCOMP1 beats DLP16 (4 cases), it generally does so with lower occupancies and higher ADRs. As for average LOS data, DCOMP1 shows about the same LOS as DCOMP in all the 10 cases (only varies by 0.01 or 0.02 days). The results also show that DCOMP1 has a slightly shorter (average of 0.04) LOS than DLP16 and slightly longer (average of 0.03) LOS than DCOMP-AVG in the first 4 cases and shorter (average of 0.04) in the next 5 cases.

**6.2.2. Computation Time** As for the computation time, the new DCOMP1 algorithm ran in 280 seconds (4.7 minutes) in Visual C++ (2010) on a PC with an Intel CORE i7 2600 CPU at 3.4 GHz, which had a Windows 7, 64-bit operating system (for comparison, Matlab took around 75 minutes to solve DCOMP1). This is a very reasonable run time, especially given the realistic size of the property (i.e., number of resources, products and time frames), which is much larger than anything previously reported in the literature. Given this computation time, a hotel or airline could easily run this every night.

### 6.3. Performance on Modified Hotel Data

We next conduct numerical simulation on modified hotel data to investigate the performance drivers of different heuristics. To this end, we vary the original data from the major hotel along two dimensions. First, we scale the data linearly so that the booking horizon, demand and capacity are all scaled down by the same factor. This variation is necessary to investigate the performance of the different policies for problems with varying capacity and booking horizon length. It is well-known that network RM problems demonstrate asymptotic behavior as they scale up (Cooper 2002). Therefore the performance difference between two strong heuristic policies may be washed away when the problem lies in the relevant asymptotic region. Second, we decrease the maximum number of nights that a guest can stay (i.e., we decrease the maximum LOS) from 12 to either 9, 6 or 2 nights. The purpose of this change is to study the implications of network structure on the performance of the various heuristics developed in the literature. The existing literature in network RM predominately considers networks with two- or three-leg itineraries (equivalent to a 2- or 3-night stay at a hotel). In the hotel’s original data set, a guest can stay up to 12 nights. Therefore the network structure is much more complex here than previously reported and we would expect to see stronger network effects.

We choose to focus on the relative performance between DCOMP1 and only two other heuristics (DCOMP, DLP16). Since the DLP policy without resolving is not competitive in most problem instances and the overall performance of DCOMP-AVG is very similar to DCOMP, we ignore DLP and DCOMP-AVG in this section. For the results from the first modification of the data (scale-down factors), Table 6 reports the revenue gain of DCOMP1 relative to DCOMP over 1000 simulations. Figure 5 shows the same information graphically. Overall, DCOMP1 shows a consistent revenue gain over DCOMP. Also, it appears that, for all scale-down factors ( $SF = 2, 5$  and  $10$ ), the gain is the highest when the demand/capacity ratio is in the intermediate range (cases #6-8), which tends to be more frequently encountered in practice. As the problem instances are scaled down linearly (e.g., move from scale factor 2 to 5), the revenue gains tend to be higher as would be

expected from the aforementioned asymptotic results. Table 7 and Figure 6 report the revenue gain of DCOMP1 compared with DLP16. Even though DCOMP1 is slightly better than DLP16 in the base case (original data, set #7), we find that DLP with frequent resolving (16 times) shows surprisingly strong performance over all the other scenarios. We conjecture that this is because the DLP formulation captures network effects very well and since this is the largest network reported in the literature to date (12 nights or connecting legs), the effect is the strongest here.

For the results of the second modification of the data (differing maximum LOS), Table 8 and Figure 7 report the revenue gain of DCOMP1 compared to DCOMP for 4 different scale-down factors and a maximum stay of 9 nights. Table 9 and Figure 8 report the revenue gain of DCOMP1 compared with DLP16. Overall, the observations are fairly consistent with the results shown earlier for a maximum stay of 12 nights (original data); that is, DCOMP1 generally beats DCOMP, but is outperformed by DLP16. We don't see any major shifts by decreasing the network size from 12 to 9 nights. Table 10 and Figure 9 report the revenue gain of DCOMP1 compared to DCOMP for 4 different scale-down factors and a maximum stay of 6 nights. Table 11 and Figure 10 report the revenue gain of DCOMP1 compared with DLP16. Overall, the observations are fairly consistent with the results shown earlier for a maximum stay of 9 or 12 nights; that is, DCOMP1 generally beats DCOMP (but the maximum revenue gain in each SF column is smaller in Table 10 than Table 8), but DCOMP1 is outperformed by DLP16. Again, we see that the absolute value of the largest revenue gain in each column is smaller in Table 11 than Table 9. Thus it appears that the network effect is now diminishing as we've dropped to 6 nights.

Table 12 and Figure 11 report the revenue gain of DCOMP1 compared to DCOMP for 4 different scale-down factors and a maximum stay of 2 nights. Table 13 and Figure 12 report the revenue gain of DCOMP1 compared with DLP16. Overall, we see that these results are the opposite of the results shown earlier for a maximum stay of 6, 9 or 12 nights; that is, DCOMP1 is now generally outperformed by DCOMP (although the maximum revenue gains are now less than 0.5% in 39 out of 40 cases), and DCOMP1 now outperforms DLP16 (revenue gains are generally less than 0.5% though).

Overall, our results for the modified hotel data show that the revenue gains from DCOMP1 can be much larger than what we reported with the original hotel data. However, the revenue gains are highly dependent on the particular problem instance. Variants of DCOMP and DLP16 are frequently used in practice. Our results show that they do perform reasonably well. However, given the relatively low implementation cost of DCOMP1, especially if DCOMP is already in place, its small but consistent revenue gains can be easily justified. One of the most interesting outcomes of

our results is the tremendous importance of network structure. Here we see that as the number of connecting legs or stay nights increases, it becomes increasingly important to have a heuristic policy that can take this into account, and thus DLP16 starts to dominate any DP heuristic that is currently mentioned in the literature.

## 7. Summary and Future Directions

This paper considers dynamic pricing for network revenue management with application to the hotel industry. Dynamic programming formulation for the problem suffers from the well-known curse of dimensionality. We generalize the nonlinear non-separable approximation developed in Zhang (2011) to the dynamic pricing setup. We show that this approach leads to a tighter upper bound on the optimal value compared with a deterministic approximation and the classical dynamic programming decomposition. Moreover, the new approach does not require significant increase in computational time. We apply the approach to data obtained from a major hotel. Simulation study based on this data shows modest revenue increase from the new approach in the base case and more significant revenue lift in many of the modified data scenarios. Also, significantly we demonstrate the key importance of network structure on the results.

Our research points to many possible avenues for future research. First, from a practical point of view, there is value to implement the approach developed in this paper. If a firm had already implemented the classical dynamic programming decomposition, the hurdle for implementing the new approach would be minimal, as it involves the same forecasts and control structure. Second, from a methodological viewpoint, there is value to explore stronger forms of functional approximations and faster methods to solve the existing approximation architecture. So far, almost all research adopting LP-based approximate dynamic programming considers models where product prices are fixed. While such models are reasonable for some industries, such as airlines, it is not suitable for industries where prices are used directly as control variables, including many hotel chains. Exploring dynamic pricing applications of LP-based approximate dynamic programming is a fruitful avenue of future research.

## References

- Adelman, D. 2007. Dynamic bid-prices in revenue management. *Operations Research* **55**(4) 647–661.
- Bertsimas, D., I. Popescu. 2003. Revenue management in a dynamic network environment. *Transportation Science* **37**(3) 257–277.
- Bitran, G., R. Caldentey. 2003. An overview of pricing models for revenue management. *Manufacturing and Service Operations Management* **5**(3) 203–229.

- 
- Bodea, T., M. Ferguson, L. Garrow. 2009. Data set—choice-based revenue management: Data from a major hotel chain. *Manufacturing and Service Operations Management* **11**(2) 356–361.
- Boyd, A., R. Kallesen. 2004. The science of revenue management when passengers purchase the lowest available fare. *Journal of Revenue and Pricing Management* **3**(2) 171–177.
- Bratu, S. 1998. Network value concept in airline revenue management. Tech. rep., Flight Transportation Laboratory, MIT.
- Chaneton, J., G. Vulcano. 2011. Computing bid prices for revenue management under customer choice behavior. *Manufacturing and Service Operations Management* **13**(4) 452–470.
- Cooper, W. L. 2002. Asymptotic behavior of an allocation policy for revenue management. *Operations Research* **50**(4) 720–727.
- Cooper, W. L., T. Homem-de-Mello. 2007. Some decomposition methods for revenue management. *Transportation Science* **41**(3) 332–353.
- Dong, L., P. Kouvelis, Z. Tian. 2009. Dynamic pricing and inventory control of substitute products. *Manufacturing and Service Operations Management* **11**(2) 317–339.
- Elmaghraby, W., P. Keskinocak. 2003. Dynamic pricing in the presence of inventory considerations: Research overview, current practices, and future directions. *Management Science* **49**(10) 1287–1309.
- Erdelyi, A., H. Topaloglu. 2011. Using decomposition methods to solve pricing problems in network revenue management. *Journal of Revenue & Pricing Management* **10**(4) 325–343.
- Farias, V. F., B. Van Roy. 2007. An approximate dynamic programming approach to network revenue management. Working paper, MIT Sloan School of Management.
- Feng, Y., G. Gallego. 1995. Optimal starting times for end-of-season sales and optimal stopping times for promotional fares. *Management Science* **41**(8) 1371–1391.
- Feng, Y., G. Gallego. 2000. Perishable asset revenue management with markovian time dependent demand intensities. *Management Science* **46**(7) 941–956.
- Feng, Y., B. Xiao. 2000. A continuous-time yield management model with multiple prices and reversible price changes. *Management Science* **46**(5) 644–657.
- Fiig, T., K. Isler, C. Hopperstad, P. Belobaba. 2009. Optimization of mixed fare structures: Theory and applications. *Journal of Revenue and Pricing Management* **9**(1) 152–170.
- Gallego, G., G. Iyengar, R. Phillips, A. Dubey. 2004. Managing flexible products on a network. CORC Technical Report Tr-2004-01, IEOR Department, Columbia University.
- Gallego, G., G. J. van Ryzin. 1994. Optimal dynamic pricing of inventories with stochastic demand over finite horizons. *Management Science* **40**(8) 999–1020.
- Gallego, G., G. J. van Ryzin. 1997. A multiproduct dynamic pricing problem and its applications to network yield management. *Operations Research* **45**(1) 24–41.

- 
- Gallego, G., R. Wang. 2014. Multi-product price optimization and competition under the nested logit model with product-differentiated price sensitivities. *Operations Research* **62**(2) 450–461.
- Goldman, P., R. Freling, K. Pak, N. Piersma. 2002. Models and techniques for hotel revenue management using a rolling horizon. *Journal of Revenue and Pricing Management* **1**(3) 207–219.
- Kleywegt, A. J. 2001. An optimal control problem of dynamic pricing. Working paper, School of Industrial Engineering, Georgia Institute of Technology.
- Kunnunkal, S., K. Talluri, H. Topaloglu. 2012. A randomized linear programming method for network revenue management with product-specific no-shows. *Transportation Science* **46**(1) 90–108.
- Kunnunkal, S., H. Topaloglu. 2008. A refined deterministic linear program for the network revenue management problem with customer choice behavior. *Naval Research Logistics* **55**(6) 563–580.
- Kunnunkal, S., H. Topaloglu. 2010a. Computing time-dependent bid prices in network revenue management problems. *Transportation Science* **44**(1) 38–62.
- Kunnunkal, S., H. Topaloglu. 2010b. A new dynamic programming decomposition method for the network revenue management problem with customer choice behavior. *Production and Operations Management* **19**(5) 575–590.
- Kunnunkal, S., H. Topaloglu. 2010c. A stochastic approximation algorithm for making pricing decisions in network revenue management problems. *Journal of Revenue & Pricing Management* **9**(5) 419–442.
- Lautenbacher, C., S. Stidham. 1999. The underlying Markov decision process in the single-leg airline yield management problem. *Transportation Science* **33**(2) 136–146.
- Li, H., W. T. Huh. 2011. Pricing multiple products with the multinomial logit and nested logit models: Concavity and implications. *Manufacturing & Service Operations Management* **13**(4) 549–563.
- Ling, L., X. Guo, L. He. 2012. Optimal pricing strategy of hotel for long-term stay. *International Journal of Services Technology and Management* **17**(1) 72–86.
- Liu, Q., G. J. van Ryzin. 2008. On the choice-based linear programming model for network revenue management. *Manufacturing and Service Operations Management* **10**(2) 288–310.
- Liu, S., K. Lai, S. Wang. 2008. Booking models for hotel revenue management considering multiple-day stays. *International Journal of Revenue Management* **2**(1) 78–91.
- Maglaras, C., J. Meissner. 2006. Dynamic pricing strategies for multi-product revenue management problems. *Manufacturing and Service Operations Management* **8**(2) 136–148.
- McGill, J. I., G. J. van Ryzin. 1999. Revenue management: Research overview and prospects. *Transportation Science* **33**(2) 233–256.
- Meissner, J., A. Strauss, K. Talluri. 2012. An enhanced concave program relaxation for choice network revenue management. Forthcoming, *Production and Operations Management*.
- Miranda Bront, J., I. Mendez-Diaz, G. Vulcano. 2009. A column generation algorithm for choice-based network revenue management. *Operations Research* **57**(3) 769–784.

- 
- Rayfield, W. Z., P. Rusmevichientong, H. Topaloglu. 2012. Approximation methods for pricing problems under the nested logit model with price bounds. Working paper, School of Operations Research and Information Engineering, Cornell University.
- Simpson, R. 1989. Using network flow techniques to find shadow prices for market and seat inventory control. Tech. rep., MIT Flight Transportation Laboratory, MIT, Cambridge, MA.
- Talluri, K., G. J. van Ryzin. 1998. An analysis of bid-price controls for network revenue management. *Management Science* **44**(11) 1577–1593.
- Talluri, K., G. J. van Ryzin. 1999. A randomized linear programming method for computing network bid prices. *Transportation Science* **33**(2) 207–216.
- Talluri, K., G. J. van Ryzin. 2004. *The Theory and Practice of Revenue Management*. Springer, New York, NY.
- Tam, W. 2008. Airline revenue management based on dynamic programming incorporating passenger sell-up behavior. MS Thesis, Massachusetts Institute of Technology.
- van Ryzin, G., G. Vulcano. 2008. Computing virtual nesting controls for network revenue management under customer choice behavior. *Manufacturing and Service Operations Management* **10**(3) 448–467.
- Vulcano, G., G. van Ryzin, W. Chaar. 2010. Choice-based revenue management: An empirical study of estimation and optimization. *Manufacturing and Service Operations Management* **12**(3) 371–392.
- Weatherford, L. 1995. Length of stay heuristics: Do they really make a difference? *Cornell Hotel and Restaurant Administration Quarterly* **36**(6) 70–79.
- Williamson, E. 1992. Airline network seat control. Ph.D. thesis, Massachusetts Institute of Technology.
- Zhang, D. 2011. An improved dynamic programming decomposition approach for network revenue management. *Manufacturing and Service Operations Management* **13**(1) 35–52.
- Zhang, D., D. Adelman. 2009. An approximate dynamic programming approach to network revenue management with customer choice. *Transportation Science* **43**(3) 381–394.
- Zhang, D., W. L. Cooper. 2005. Revenue management for parallel flights with customer-choice behavior. *Operations Research* **53**(3) 415–431.
- Zhang, D., W. L. Cooper. 2009. Pricing substitutable flights in airline revenue management. *European Journal of Operational Research* **197**(3) 848–861.
- Zhang, D., Z. Lu. 2013. Assessing the value of dynamic pricing in network revenue management. *INFORMS Journal on Computing* **25**(1) 102–115.
- Zhao, W., Y.-S. Zheng. 2000. Optimal dynamic pricing for perishable assets with nonhomogeneous demand. *Management Science* **46**(3) 375–388.

Case #	Capacity per stay night	Demand/capacity ratio
1	80	2.53
2	100	2.02
3	120	1.69
4	140	1.44
5	160	1.26
6	180	1.12
7	200	1.01
8	220	0.92
9	240	0.84
10	260	0.78

**Table 1** Description of test cases

Case #	DLP bound	DCOMP bound	DCOMP1 bound	Bound improvement	
				%-DLP	%-DCOMP
1	2400.00	2398.60	2398.60	0.06%	0.00%
2	2970.24	2967.82	2968.06	0.07%	0.01%
3	3508.72	3505.61	3506.05	0.08%	0.01%
4	4011.09	4006.95	4007.27	0.10%	0.01%
5	4458.25	4452.31	4452.51	0.13%	0.00%
6	4824.55	4815.28	4815.78	0.18%	0.01%
7	5039.65	5029.70	5030.28	0.19%	0.01%
8	5143.55	5134.50	5135.39	0.16%	0.02%
9	5189.12	5178.92	5179.32	0.19%	0.01%
10	5196.95	5193.66	5193.68	0.06%	0.00%

**Table 2** Percentage improvement of DCOMP1's upper bound over DLP and DCOMP

Case #	% DLP	% DLP16	%DCOMP	%DCOMP-AVG
1	2.00%	-0.73%	-0.01%	0.48%
2	3.25%	-0.30%	-0.20%	0.12%
3	2.93%	-0.08%	-0.09%	0.32%
4	2.33%	-0.07%	0.00%	0.30%
5	1.77%	-0.08%	0.01%	0.25%
6	1.36%	-0.10%	0.15%	0.12%
<b>7</b>	<b>1.10%</b>	<b>0.01%</b>	<b>0.14%</b>	<b>0.13%</b>
8	0.85%	0.05%	0.05%	0.09%
9	0.36%	0.03%	0.04%	0.04%
10	0.10%	0.00%	0.00%	0.01%

**Table 3** Simulated average percentage revenue gains of DCOMP1 over 4 different control policies

Case #	DLP16	DCOMP	DCOMP AVG	DCOMP1
1	98.80%	98.40%	97.60%	98.40%
2	98.70%	98.70%	98.10%	98.40%
3	98.60%	98.60%	97.80%	98.40%
4	98.40%	98.30%	97.50%	98.20%
5	98.10%	97.80%	97.10%	97.60%
6	97.40%	96.40%	96.60%	96.70%
<b>7</b>	<b>95.30%</b>	<b>94.00%</b>	<b>94.80%</b>	<b>94.80%</b>
8	91.00%	90.20%	90.60%	90.80%
9	85.70%	85.20%	85.50%	85.70%
10	79.90%	79.80%	79.90%	79.70%

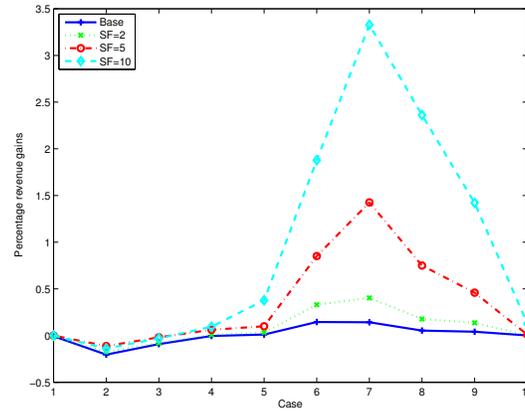
**Table 4** Empirical average occupancy for different control policies

Case #	DLP16		DCOMP		DCOMP-AVG		DCOMP1	
	Average rate/night	Average LOS						
1	\$149.40	4.44	\$148.91	4.19	\$149.34	4.13	\$148.91	4.19
2	\$147.64	4.35	\$147.57	4.30	\$147.95	4.27	\$147.62	4.31
3	\$145.09	4.36	\$145.10	4.36	\$145.63	4.31	\$145.24	4.34
4	\$142.04	4.38	\$142.06	4.38	\$142.76	4.37	\$142.24	4.37
5	\$138.20	4.40	\$138.53	4.38	\$139.16	4.41	\$138.71	4.37
6	\$133.81	4.40	\$134.87	4.35	\$134.74	4.44	\$134.67	4.37
<b>7</b>	<b>\$129.68</b>	<b>4.35</b>	<b>\$131.22</b>	<b>4.32</b>	<b>\$130.25</b>	<b>4.39</b>	<b>\$130.33</b>	<b>4.33</b>
8	\$127.05	4.32	\$128.11	4.30	\$127.51	4.34	\$127.34	4.31
9	\$125.43	4.30	\$126.19	4.29	\$125.74	4.31	\$125.42	4.29
10	\$124.80	4.29	\$125.00	4.29	\$124.86	4.29	\$125.17	4.29

**Table 5** Average per-night rates and LOS for accepted reservations in policy simulations

Case #	Base case	Scale down by 2	Scale down by 5	Scale down by 10
1	-0.01%	-0.01%	0.00%	0.00%
2	-0.20%	-0.17%	-0.11%	-0.14%
3	-0.09%	-0.08%	-0.02%	-0.03%
4	0.00%	0.01%	0.06%	0.09%
5	0.01%	0.03%	0.10%	0.38%
6	0.15%	0.33%	0.85%	1.88%
<b>7</b>	<b>0.14%</b>	<b>0.40%</b>	<b>1.43%</b>	<b>3.33%</b>
8	0.05%	0.18%	0.75%	2.36%
9	0.04%	0.14%	0.46%	1.42%
10	0.00%	0.00%	0.01%	0.13%

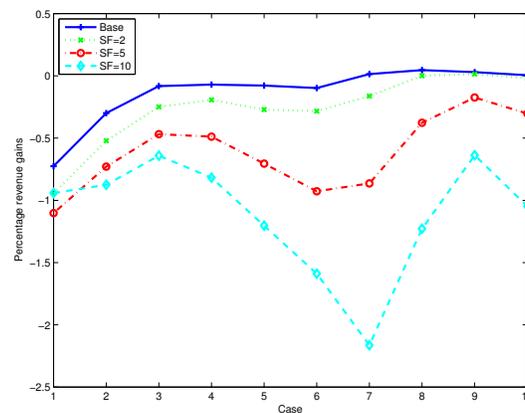
**Table 6** Percentage revenue gains of DCOMP1 compared with DCOMP for different scale-down factors with maximum of 12 stay nights



**Figure 5** Percentage revenue gains of DCOMP1 compared with DCOMP for different scale-down factors with maximum of 12 stay nights

Case #	Base case	Scale down by 2	Scale down by 5	Scale down by 10
1	-0.73%	-0.94%	-1.10%	-0.94%
2	-0.30%	-0.52%	-0.73%	-0.88%
3	-0.08%	-0.25%	-0.47%	-0.64%
4	-0.07%	-0.19%	-0.49%	-0.82%
5	-0.08%	-0.27%	-0.70%	-1.20%
6	-0.10%	-0.28%	-0.93%	-1.59%
7	<b>0.01%</b>	<b>-0.16%</b>	<b>-0.86%</b>	<b>-2.16%</b>
8	0.05%	0.00%	-0.38%	-1.23%
9	0.03%	0.01%	-0.17%	-0.64%
10	0.00%	-0.02%	-0.30%	-1.05%

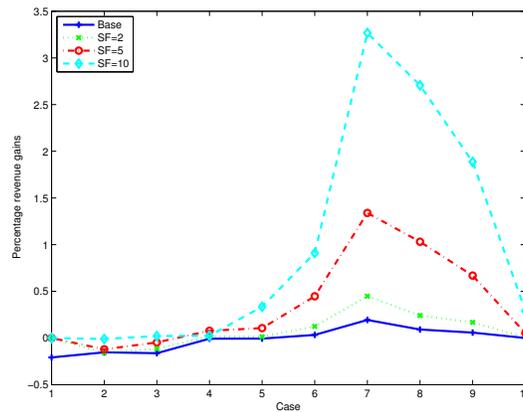
**Table 7** Percentage revenue gains of DCOMP1 compared with DLP16 for different scale-down factors with maximum of 12 stay nights



**Figure 6** Percentage revenue gains of DCOMP1 compared with DLP16 for different scale-down factors with maximum of 12 stay nights

Case #	Base case	Scale down by 2	Scale down by 5	Scale down by 10
1	-0.21%	0.00%	0.00%	0.00%
2	-0.15%	-0.16%	-0.12%	-0.01%
3	-0.16%	-0.12%	-0.05%	0.02%
4	-0.01%	0.02%	0.08%	0.02%
5	-0.01%	0.01%	0.11%	0.34%
6	0.03%	0.12%	0.45%	0.91%
7	<b>0.19%</b>	<b>0.45%</b>	<b>1.34%</b>	<b>3.27%</b>
8	0.09%	0.24%	1.03%	2.71%
9	0.06%	0.17%	0.67%	1.89%
10	0.00%	0.00%	0.05%	0.29%

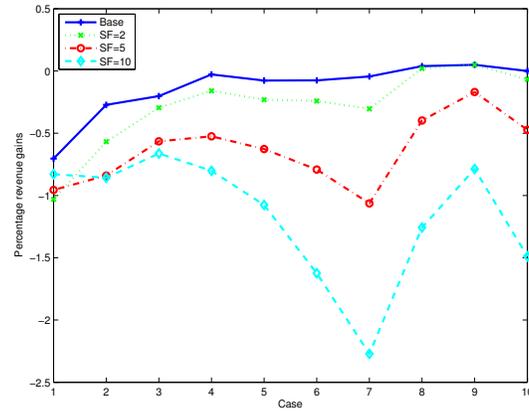
**Table 8** Percentage revenue gains of DCOMP1 compared with DCOMP for different scale-down factors with maximum of 9 stay nights



**Figure 7** Percentage revenue gains of DCOMP1 compared with DCOMP for different scale-down factors with maximum of 9 stay nights

Case #	Base case	Scale down by 2	Scale down by 5	Scale down by 10
1	-0.70%	-1.03%	-0.96%	-0.83%
2	-0.27%	-0.57%	-0.84%	-0.86%
3	-0.20%	-0.30%	-0.57%	-0.66%
4	-0.03%	-0.16%	-0.52%	-0.80%
5	-0.08%	-0.23%	-0.63%	-1.08%
6	-0.08%	-0.24%	-0.79%	-1.62%
7	<b>-0.04%</b>	<b>-0.30%</b>	<b>-1.06%</b>	<b>-2.27%</b>
8	0.04%	0.02%	-0.40%	-1.26%
9	0.05%	0.05%	-0.17%	-0.79%
10	0.00%	-0.07%	-0.47%	-1.49%

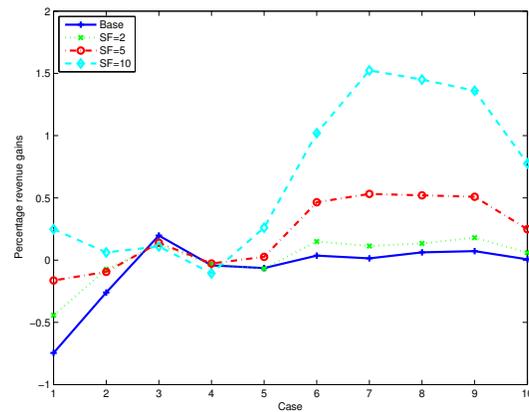
**Table 9** Percentage revenue gains of DCOMP1 compared with DLP16 for different scale-down factors with maximum of 9 stay nights



**Figure 8** Percentage revenue gains of DCOMP1 compared with DLP16 for different scale-down factors with maximum of 9 stay nights

Case #	Base case	Scale down by 2	Scale down by 5	Scale down by 10
1	-0.75%	-0.44%	-0.16%	0.25%
2	-0.26%	-0.07%	-0.09%	0.06%
3	0.20%	0.12%	0.14%	0.11%
4	-0.04%	-0.03%	-0.03%	-0.11%
5	-0.06%	-0.07%	0.03%	0.26%
6	0.04%	0.15%	0.46%	1.02%
<b>7</b>	<b>0.01%</b>	<b>0.11%</b>	<b>0.53%</b>	<b>1.52%</b>
8	0.06%	0.13%	0.52%	1.45%
9	0.07%	0.18%	0.51%	1.36%
10	0.01%	0.06%	0.25%	0.78%

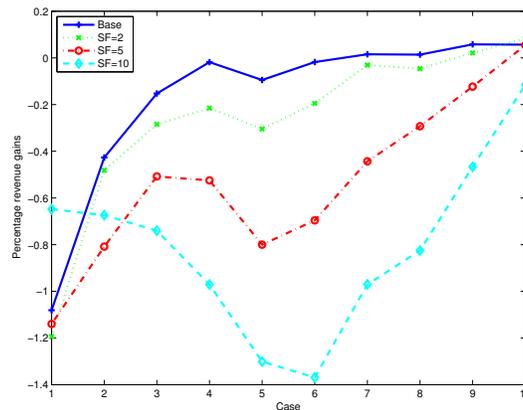
**Table 10** Percentage revenue gains of DCOMP1 compared with DCOMP for different scale-down factors with maximum of 6 stay nights



**Figure 9** Percentage revenue gains of DCOMP1 compared with DCOMP for different scale-down factors with maximum of 6 stay nights

Case #	Base case	Scale down by 2	Scale down by 5	Scale down by 10
1	-1.08%	-1.19%	-1.14%	-0.65%
2	-0.43%	-0.48%	-0.81%	-0.67%
3	-0.15%	-0.28%	-0.51%	-0.74%
4	-0.02%	-0.22%	-0.52%	-0.97%
5	-0.09%	-0.30%	-0.80%	-1.30%
6	-0.02%	-0.19%	-0.70%	-1.37%
7	<b>0.02%</b>	<b>-0.03%</b>	<b>-0.44%</b>	<b>-0.97%</b>
8	0.01%	-0.05%	-0.29%	-0.83%
9	0.06%	0.02%	-0.12%	-0.47%
10	0.06%	0.09%	0.05%	-0.12%

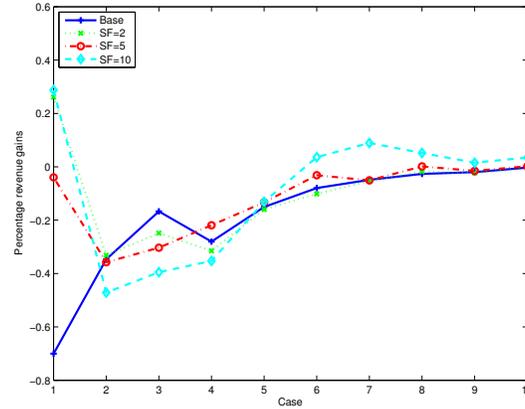
**Table 11** Percentage revenue gains of DCOMP1 compared with DLP16 for different scale-down factors with maximum of 6 stay nights



**Figure 10** Percentage revenue gains of DCOMP1 compared with DLP16 for different scale-down factors with maximum of 6 stay nights

Case #	Base case	Scale down by 2	Scale down by 5	Scale down by 10
1	-0.70%	0.26%	-0.04%	0.29%
2	-0.35%	-0.33%	-0.36%	-0.47%
3	-0.17%	-0.25%	-0.30%	-0.40%
4	-0.28%	-0.31%	-0.22%	-0.35%
5	-0.15%	-0.16%	-0.13%	-0.13%
6	-0.08%	-0.10%	-0.03%	0.04%
7	<b>-0.05%</b>	<b>-0.05%</b>	<b>-0.05%</b>	<b>0.09%</b>
8	-0.03%	-0.02%	0.00%	0.05%
9	-0.02%	-0.02%	-0.02%	0.02%
10	0.00%	-0.01%	0.00%	0.03%

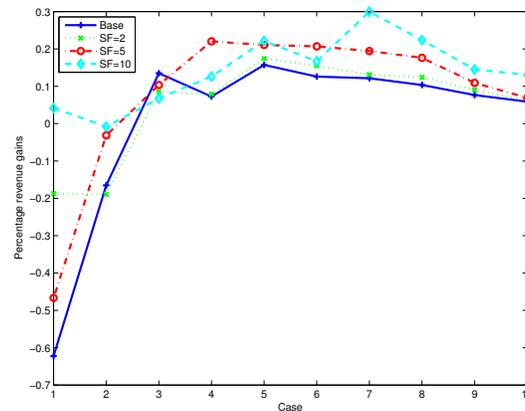
**Table 12** Percentage revenue gains of DCOMP1 compared with DCOMP for different scale-down factors with maximum of 2 stay nights



**Figure 11** Percentage revenue gains of DCOMP1 compared with DCOMP for different scale-down factors with maximum of 2 stay nights

Case #	Base case	Scale down by 2	Scale down by 5	Scale down by 10
1	-0.62%	-0.19%	-0.47%	0.04%
2	-0.17%	-0.19%	-0.03%	-0.01%
3	0.13%	0.09%	0.10%	0.07%
4	0.07%	0.08%	0.22%	0.13%
5	0.16%	0.17%	0.21%	0.22%
6	0.13%	0.16%	0.21%	0.17%
7	<b>0.12%</b>	<b>0.13%</b>	<b>0.19%</b>	<b>0.30%</b>
8	0.10%	0.12%	0.18%	0.22%
9	0.08%	0.09%	0.11%	0.15%
10	0.06%	0.06%	0.07%	0.13%

**Table 13** Percentage revenue gains of DCOMP1 compared with DLP16 for different scale-down factors with maximum of 2 stay nights



**Figure 12** Percentage revenue gains of DCOMP1 compared with DLP16 for different scale-down factors with maximum of 2 stay nights