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. This paper considers a dynamic programming formulation for the problem. Due to the well-known curse of dimensionality, solving the problem exactly is out of reach. We generalize the non-linear non-separable approximation architecture proposed in Zhang (2011) to tackle this problem. We show that the new approach generates an upper bound that is tighter than the bounds from a deterministic approximation and a dynamic programming decomposition scheme. We apply the new approach to problem instances generated based on the data obtained from a major resort hotel in the US. A numerical study shows that a heuristic control policy based on the new approach leads to higher revenues than several benchmark policies.

This version: December 12, 2012

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. Even though DCOMP has been shown to improve expected revenue
performance over DLP, a critical deficiency of the approach is that by decomposing the problem into single resource problems, important network information is lost.

One way to improve DCOMP is to look for ways to better incorporate network information. Zhang (2011) proposes an improvement to DCOMP by considering a non-linear non-separable functional approximation to dynamic programming value functions. Since the functional approximation 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 (2011) 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 (2011). We show that applying a nonlinear non-separable functional approximation leads to an improved bound compared with the network decomposition in Zhang and Lu (2011). This result is a generalization of the results in Zhang (2011).

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. Typically, rolling-horizon procedures are used to solve the problem at a given cut-off date. Such a formulation is indeed considered by many practical RM systems, including the one used at our partner firm. 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 resort hotel in the US. 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 we are aware of in the public domain. There is substantial revenue potential from the new approach (e.g., a 1% revenue gain for an airline/hotel company with annual revenues of $10 billion would be $100 million). Compared with existing work, the benefit of the proposed approach lies both in its revenue enhancement over leg DP and its relatively low computational cost. Since the hotel setting is different (Vinod 2004) from the airline problem that motivated the original Zhang (2011) study, we will: (a) first generalize the new approach to the hotel context, and (b) test the computational and policy performance of the new approach on the actual hotel data.

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 resort hotel in the US. Section 5 describes control policies from the three alternative solution strategies. Section 6 reports numerical results on the revenue performance of our new algorithm and 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 stochastic LP outperformed deterministic LP by 2-3.5% in a simulated data set. 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

---

1. 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.
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.

2. Model Formulation

We consider the dynamic pricing problem in a network with $m$ resources. The network capacity is denoted by a vector $c = (c_1, \ldots, c_m)$, where $c_i$ is the capacity of resource $i$; $i = 1, \ldots, m$. 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:

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

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

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

In the above, $R_t(x) = \times_{j=1}^n R_{t,j}(x)$, where $R_{t,j}(x) = \mathbb{R}_+$ 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

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

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 2011)

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

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

**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 (2011) 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

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

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

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

The boundary conditions are

\[
v_{T+1,i}(x) = 0, \quad \forall x, \\
v_{t,i}(0) = 0, \quad \forall t.
\]

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 (2011) 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 (2011)). For each \( i \), let \( v_{t,i}^*(\cdot) \) be an optimal solution and a feasible solution to (LP\(_i\)). Then, for each \( k = 1, \ldots, m \), 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
\[
v_t(x) \approx \min_i \left\{ \hat{v}_{t,i}(x_i) + \sum_{k \neq i} \pi_k^* x_k \right\}, \quad \forall t, x.
\]
(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
\[
\text{(NLP)} \quad z_{NLP} = \min_{\hat{v}_{t,i}(\cdot)_{\forall t,i}} \min_i \left\{ \hat{v}_{1,i}(c_i) + \sum_{k \neq i} \pi_k^* c_k \right\}
\]
\[
\min_i \left\{ \hat{v}_{t,i}(x_i) + \sum_{k \neq i} \pi_k^* x_k \right\} \geq \min_{\lambda} \left\{ \sum_{j=1}^m \lambda P_j(r_t) \left( r_{t,j} + \min_l \left\{ \hat{v}_{t+1,l}(x_l - a_{lj}) + \sum_{k \neq l} \pi_k^* (x_k - a_{kj}) \right\} \right) \right\}
\]
\[
+ (\lambda P_0(r_t) + 1 - \lambda) \min_i \left\{ \hat{v}_{t+1,i}(x_l) + \sum_{k \neq l} \pi_k^* x_k \right\}, \quad \forall t, x, r_t \in R_t(x).
\]
(4)

First, observe that the minimization on the left-hand side of (4) can be removed by writing each constraint as \( m \) equivalent constraints
\[
\hat{v}_{t,i}(x_i) + \sum_{k \neq i} \pi_k^* x_k \geq \sum_{j=1}^m \lambda P_j(r_t) \left( r_{t,j} + \min_{\lambda} \left\{ \hat{v}_{t+1,j}(x_l - a_{lj}) + \sum_{k \neq l} \pi_k^* (x_k - a_{kj}) \right\} \right)
\]
\[
+ (\lambda P_0(r_t) + 1 - \lambda) \min_{\lambda} \left\{ \hat{v}_{t+1,j}(x_l) + \sum_{k \neq l} \pi_k^* x_k \right\}, \quad \forall i, t, x, r_t \in R_t(x).
\]
(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

$$
\hat{v}_{t,i}(x_i) \geq \sum_{j=1}^{n} \lambda P_j(r_t) \left( r_{t,j} + \min_{l} \left\{ \hat{v}_{t+1,l}(x_i - a_{ij}) + \sum_{k \neq l} \pi^*_k (x_k - a_{kj}) - \sum_{k \neq l} \pi^*_k x_k \right\} \right)
$$

$$
+ (\lambda P_0(r_t) + 1 - \lambda) \min_{l} \left\{ \hat{v}_{t+1,l}(x_i) + \sum_{k \neq l} \pi^*_k x_k - \sum_{k \neq l} \pi^*_k x_k \right\}, \forall i, t, x, r_t \in R_t(x).
$$

Simplifying the above leads to

$$
\hat{v}_{t,i}(x_i) \geq \sum_{j=1}^{n} \lambda P_j(S) \left( r_{t,j} + \min_{l} \left\{ \hat{v}_{t+1,l}(x_i - a_{ij}) - \sum_{k \neq l} \pi^*_k a_{kj}, \right\} \right)
$$

$$
\min_{l \neq l} \left\{ \hat{v}_{t+1,l}(x_i - a_{ij}) - (x_i - a_{ij}) \pi_i^* - \sum_{k} a_{kj} \pi_k^* + \pi_i^* x_i \right\}
$$

$$
+ (\lambda P_0(r_t) + 1 - \lambda) \min_{l} \left\{ \hat{v}_{t+1,l}(x_i), \min_{l \neq l} \left\{ \hat{v}_{t+1,l}(x_i) - \pi_i^* x_i + \pi_i^* x_i \right\} \right\}, \forall i, t, x, r_t \in R_t(x).
$$

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

$$
\hat{v}_{t,i}(x_i) \geq \sum_{j=1}^{n} \lambda P_j(r_t) \left( r_{t,j} + \min_{l} \left\{ \hat{v}_{t+1,l}(x_i - a_{ij}) - \sum_{k \neq l} \pi^*_k a_{kj}, \right\} \right)
$$

$$
\min_{l \neq l} \left\{ \hat{v}_{t+1,l}(x_i - a_{ij}) - (x_i - a_{ij}) \pi_i^* - \sum_{k} a_{kj} \pi_k^* + \pi_i^* x_i \right\}
$$

$$
+ (\lambda P_0(r_t) + 1 - \lambda) \min_{l} \left\{ \hat{v}_{t+1,l}(x_i), \min_{l \neq l} \left\{ \hat{v}_{t+1,l}(x_i) - \pi_i^* x_i + \pi_i^* x_i \right\} \right\}, \forall i, t, x, r_t \in R_t(x).
$$

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

$$
(NLP) \quad z_{NL} = \min_{\hat{v}_{t,i}(c_i) \forall i} \min_{l \neq i} \left\{ \hat{v}_{t,i}(c_i) + \sum_{k \neq i} \pi^*_k c_k \right\}
$$
Proposition 4. For each feasible solution \( \{\hat{v}_{t,i}(x_i)\}_{t,i} \) of \((\text{NLP})\), there exists a feasible solution with equal or smaller objective value such that equality in (6) holds for some \( \hat{r}_{t,i}(x_i) \in R_t(x_i, c_i) \) for each \( t, i, x_i \).

Proof. Let \( \{\hat{v}_{t,i}(\cdot)\}_{t,i} \) be a feasible solution to \((\text{NLP})\). 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 \( \{\hat{v}_{t,i}\}_{t,i} \) can be modified by replacing \( \hat{v}_{t,i}(x_i) \) with

\[
v^+_{t,i}(x_i) = \max_{r_t \subseteq N(x_i, c_i)} \sum_{j=1}^n \lambda P_j(r_t) \left( r_{t,j} + \min_{l \neq i} \left\{ \max_{0 \leq y_l \leq c_l - a_{lj}} [\hat{v}_{t+1,l}(y_l) - y_l \pi^*_l] - \sum_k a_{kj} \pi^*_k + \pi^*_i x_i \right\} \right)
\]

\[
+ (\lambda P_0(r_t) + 1 - \lambda) \min_{l \neq i} \left\{ \max_{0 \leq y_l \leq c_l - a_{lj}} [\hat{v}_{t+1,l}(y_l) - y_l \pi^*_l] - \sum_k a_{kj} \pi^*_k + \pi^*_i x_i \right\} .
\]

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.

An immediate corollary for Proposition 4 is the following.

Corollary 1. There exists an optimal solution \( \{\hat{v}_{t,i}(\cdot)\}_{t,i} \) to \((\text{NLP})\) such that equality holds for some \( \hat{r}_{t,i}(x_i) \in R_t(x_i, c_i) \) for each fixed \( t, i, x_i \).
Observe that, by construction, each constraint in (\(\bar{\text{NLP}}\)) only involves \(x_i\) for one resource \(i\), but not for all other resources. Define a set of value functions \(\{\hat{v}_{t,i}(\cdot)\}\) for all \(t\) and \(i\) as follows:

\[
\begin{align*}
\hat{v}_{t,i}(x_i) &= \max_{r_t \in R_t(x_i, c_i, -)} \sum_{j=1}^{n} \lambda P_j(r_t) \left( r_{t,j} + \min_{l \neq j} \left\{ \hat{v}_{t+1,i}(x_i - a_{ij}) - \sum_{k \neq i} \pi_k^* a_{kj} \right\} \right) \\
&\quad \min_{l \neq i} \left\{ \max_{0 \leq y_l \leq c_{l-i}} \left[ \hat{v}_{t+1,i}(y_l) - y_l \pi_i^* \right] - \sum_{k} a_{kj} \pi_k^* + \pi_i^* x_i \right\} \\
&\quad + (\lambda P_0(r_t) + 1 - \lambda) \min_{l \neq i} \left\{ \max_{0 \leq y_l \leq c_{l-i}} \left[ \hat{v}_{t+1,i}(y_l) - \pi_i^* y_l \right] + \pi_i^* x_i \right\}, \\
&\quad \forall i, t, x_i,
\end{align*}
\]

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

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

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

Proof. From Corollary 1, it is without loss of optimality to restrict our attention in (\(\bar{\text{NLP}}\)) 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 (\(\bar{\text{NLP}}\)). Then,

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

Comparing (7) with (\(\bar{\text{DP}}\)) yields the result.

Because of the equivalence of \(\hat{v}_{t,i}(x_i)\) in (\(\bar{\text{DP}}\)) 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 (\(\bar{\text{DP}}\)) instead of (\(\bar{\text{NLP}}\)). The formulation (\(\bar{\text{DP}}\)) 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 \(x_k\) and \(k = 1, \ldots, m\).

We show that the bounds from (\(\bar{\text{NLP}}\)) are tighter than the decomposition bound in Proposition 2 originally shown in Zhang and Lu (2011).

**Proposition 6.** The following results hold:

(i) \(\hat{v}_{t,i}(x_i) \leq \hat{v}_{t,i}^*(x_i), \forall i, x_i;\)

(ii) \(v_1(c) \leq z_{\bar{\text{NLP}}} \leq z_{\bar{\text{DP}}} = \min_i \left\{ \hat{v}_{t,i}^*(c_i) + \sum_{k \neq i} \pi_k^* c_k \right\} \leq \min_i \left\{ v_{t,i}(c_i) + \sum_{k \neq i} \pi_k^* c_k \right\} \leq z_{\bar{\text{DM}}}.\)
4. **Case Study at a Major US Hotel**

We test the approach proposed in this paper on real hotel data (i.e., a large US 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]). 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 would need to be considered.

---

2 Due to a confidentiality agreement with our industry partner, we have to suppress some details when reporting the data.
Figure 2  Percentage of demand by booking period

Figure 3  Percentage of demand by length-of-stay (LOS)

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.

Figure 3 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 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.
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. In discussion with our industry partner, we confirmed that this handling of demand closely resembled the approach used in practice.

Even though we develop theoretical results for a rather general demand model, the demand model used at the resort hotel is fairly specific, and corresponds to what is known in the literature as *priceable demand* (Boyd and Kallesen 2004).

We use the following linear programming formulation:

\[
\text{(LP)} \quad \max_x \sum_{i,l} \sum_{k} l f_k x_{ilk} \\
\sum_{i,l} \sum_{k} \sum_{i'=i-11}^{i+12} x_{i'lk} \leq c_i, \quad \forall i, \quad (8) \\
\sum_{k} x_{ilk} \leq \sum_{k'} d_{ilk'}, \quad \forall i,l,k, \quad (9)
\]
The formulation (LP) is a close cousin of the widely used deterministic LP formulation for network revenue management with independent demand. Here, \( f_k \) is the per night room rate in rate category \( k \). We assume the rate categories are ranked from high to low, so that \( f_1 > f_2 > f_3 \), etc. The decision variable \( x_{ilk} \) is the number of reservations for room-night \( i \) with length of stay \( l \) and price level \( k \). Note that the rate for an \( l \)-night stay is exactly \( l \) times the per night room rate \( f_k \) for a given price level \( k \). The constraint (8) is a resource constraint. The demand constraint (9) takes into account priceable demand by requiring that total number of accepted reservations above category \( k \) (i.e., categories 1 to \( k \)) is less than the total demand above category \( k \).

5. Control Policies

In this section, we introduce heuristic policies from (DP). For comparison purposes, we also consider policies from (LP) and the classical dynamic programming decomposition (Zhang and Lu 2011).

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 \( \Delta_j v_t(x) \) 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 \( p_j > \sum_i a_{ij} \pi_i^* \) 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 \( \{v_{t,i}(\cdot)\}_{v_{t,i}} \) is computed, the value function \( v_t(x) \) can then be approximated by

\[
v_t(x) \approx \sum_{i=1}^{m} v_{t,i}(x_i).
\]
By using (10), we have
\[
\Delta_j v_t(x) = v_t(x) - v_t(x - A^j) \equiv \sum_{i=1}^{m} \Delta_j v_{t,i}(x). 
\]

An approximate policy to (DP) is given by
\[
r^*_t(x) = \arg\max_{r_t \in R_t(x)} \left\{ \sum_{j=1}^{n} \lambda P_j(r_t) \left[ r_{t,j} - \sum_{i=1}^{m} \Delta_j v_{t+1,i}(x_i) \right] \right\}. 
\]

5.3. DCOMP1

In this subsection, we introduce a heuristic policy from the solution of (DP). Recall that the solution is denoted by \( \{ \hat{v}^*_{t,i}(\cdot) \}_{v_t,i} \). 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
\[
\Delta_j v_{t+1}(x) = v_{t+1}(x) - v_{t+1}(x - A^j) \approx \sum_i \hat{v}^*_{t+1,i}(x_i) - \sum_i \hat{v}^*_{t+1,i}(x_i - a_{ij}) \equiv \Delta_j v_{t+1}(x). 
\]

By replacing \( \Delta_j v_{t+1}(x) \) in (DP) with \( \Delta_j v_{t+1}(x) \) for all \( j \), the prices in period \( t \) are given by
\[
r^*_t(x) = \arg\max_{r_t \in R_t(x)} \left\{ \sum_{j=1}^{n} \lambda P_j(r_t) \left[ r_{t,j} - \sum_{i=1}^{m} \Delta_j v_{t+1,i}(x_i) \right] \right\}. 
\]

A heuristic policy that offers prices \( r^*_t(x) \) 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 5.3. The performance of DCOMP1 is compared to three other policies:

- DLP: the bid-price policy based on the dual values of resource constraints in the LP formulation (Section 5.1);
- DLP16: a version of DLP that re-solves 16 times before the stay night, with equally spaced resolving intervals;
- DCOMP: the classical dynamic programming decomposition (Section 5.2).

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
\[ \rho = \frac{\sum_t \sum_j \sum_i a_{ij} \lambda_{t,j}}{\sum_i c_i}. \]

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. First, note that both decomposition bounds are much
better (i.e., tighter) than the DLP bound. In particular, the DCOMP1 bound is 12.3% smaller
than the DLP bound in case #10. Furthermore the DCOMP1 bound is tighter than the DCOMP
bound, confirming our theoretical results in Section 3. On average, across the ten cases, DCOMP1
bounds are 4.41% and 0.004% tighter than the DLP and DCOMP bounds respectively. Note also
that the DCOMP/DCOMP1 bounds are the minimum values across all 30 stay nights for each
case.

6.2. Policy Performance of Algorithms on Original Data

The revenue performance of the four different policies is evaluated through simulation. In our
computer implementation, a uniform [0,1] random variable is generated for each time period. The
arrival in period \( t \) is classified a class-\( j \) arrival if the value of the random variable is between
\( \sum_{k=1}^{j-1} \lambda_{t,j} \) and \( \sum_{k=1}^{j} \lambda_{t,j} \). We randomly generated 5,000 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 four simulated policies in each case.

6.2.1. Revenue Results from Ten Cases Table 3 reports the average revenue results
for the ten cases considered in our simulation. The first four columns contain the revenues and
are reported as the averages over 5,000 simulations. The DCOMP1 bound column reports the
revenue upper bounds from Proposition 6. The OPT-GAP column reports the optimality gap of
the DCOMP1 policy, which is calculated as
\[ \frac{\text{DCOMP1 REV} - \text{DCOMP1 bound}}{\text{DCOMP1 bound}} \times 100\%. \]
The results show that DCOMP1 performs quite well with an optimality gap of 5.1% or less in all cases and less than 3.5% in half the cases. The last three columns measure the percentage revenue gains of DCOMP1 against three standard competitors—DLP, DLP16, and DCOMP. The results show that DCOMP1 outperforms DLP by as much as 7.46% and an average of 4.5% across the ten cases. They also show that DCOMP1 outperforms a frequently-resolved DLP algorithm (DLP16) and DCOMP by as much as 1.06% and 0.21% respectively. Although it may not seem like much, this truly is a significant improvement, especially given that DP decomposition is the state-of-the-art in the industry in terms of implemented algorithms. For a $10 billion company, this would represent an increase in revenue of $21 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., cases 9 and 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 (by 0.1-0.4%) over most of the ten cases. As we will see in Table 5, DCOMP1 is able to squeeze out a higher average daily room rate to more than compensate for the lost occupancy and thus earn more revenue overall. Compared to DLP, DCOMP1 generally has much higher average occupancy percentages (e.g., 4.5 - 9.8% higher in the majority of the cases).

Finally, Table 5 reports both the 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 ADR (e.g., $0.2 to $0.8) than DCOMP in all but one of the ten cases. DCOMP1 also has a lower ADR than DLP, but again this was more than compensated for by much higher average occupancies as seen in Table 4. As for average LOS data, DCOMP1 shows a slightly shorter LOS than DCOMP in the majority of the cases. Additionally, DCOMP1 has slightly shorter average LOS compared to DLP in the first case and slightly longer LOS in cases 3–6.

6.2.2. Computation Time As for the computation time, the new DCOMPl 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 performance drivers of different heuristics. To this end, we vary the original data from our industry partner along two dimensions. First, we decrease the number of rate categories from five to two. This change is based on the observation that the performance gap among different policies is often driven by *price differentials* among different rate categories. Having more than two rate categories will only obscure our investigation. We then redistribute demand such that 20% of the total demand is for rate category 1 and the remaining demand is for rate category 2. The rate for rate category 2 is set to be 40% of rate category 1. Note that by changing the data in this fashion, we have kept other aspects of the data, such as demand/capacity ratio and demand composition across different lengths-of-stay and different booking periods unchanged.

Second, we scale the data linearly so that the booking horizon and capacity are both 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 revenue management 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 regime.

We choose to focus on the relative performance between DCOMP1 and other heuristics. Since the DLP policy without resolving is not competitive in most problem instances, we ignore DLP policy in this section. Table 6 reports the revenue gain of DCOMP1 relative to DCOMP over 5000 simulations. Figure 4 shows the same information graphically. We also added the data series demand/capacity ratio in this figure. Overall, DCOMP1 shows a small but consistent revenue gain over DCOMP. Also, it appears that, for scale down factors 5 and 10, the gain is the highest when demand/capacity ratio is in the intermediate range, which tends to be more frequently encountered in practice. As the problem instances are scaled down linearly, the revenue gains tend to be higher. This is expected from the aforementioned asymptotic results.

Table 7 and Figure 5 show the revenue gains of DCOMP1 compared with DLP16. Overall, the observations are consistent with the comparison between DCOMP1 and DCOMP, although the magnitude of revenue gains tends to be larger, which can be attributed to the stronger performance of DCOMP compared with DLP16. One difference here is that, unlike the comparison with DCOMP, the gains are not increasing in the scale-down factor. This can at least be partially attributed to the relatively higher resolving frequency of DLP16 when the scale-down factor is larger (since there are fewer periods for instances with higher scale-down factors).
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. Our numerical results here are consistent with numerical results reported in Zhang (2011) for simulated airline data. However, the revenue gains are highly dependent on the particular problem instances. 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.

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 resort hotel in the US. Simulation study based on this data shows significant revenue increase from the new approach.

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. We are in discussion with our partner firm for potential implementation. Since the firm has already implemented the classical dynamic programming decomposition, the hurdle for implementing the new approach is 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


Weatherford, L.R. 1995. Length of stay heuristics: Do they really make a difference? *Cornell Hotel and Restaurant Administration Quarterly* 36(6) 70–79.


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

**Table 1**  Description of test cases

<table>
<thead>
<tr>
<th>Case #</th>
<th>DLP bound</th>
<th>DCOMP bound</th>
<th>DCOMP1 bound</th>
<th>Bound improvement</th>
<th>%-%-DLP</th>
<th>%-%-DCOMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>356577</td>
<td>355980</td>
<td>355953</td>
<td>0.17%</td>
<td>0.0075%</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>440478</td>
<td>438423</td>
<td>438413</td>
<td>0.47%</td>
<td>0.0024%</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>521472</td>
<td>515051</td>
<td>515024</td>
<td>1.24%</td>
<td>0.0051%</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>594315</td>
<td>581376</td>
<td>581352</td>
<td>2.18%</td>
<td>0.0041%</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>654903</td>
<td>626624</td>
<td>626621</td>
<td>4.32%</td>
<td>0.0004%</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>701175</td>
<td>659086</td>
<td>659056</td>
<td>6.01%</td>
<td>0.0046%</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>739256</td>
<td>686715</td>
<td>686684</td>
<td>7.11%</td>
<td>0.0046%</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>767297</td>
<td>698580</td>
<td>698576</td>
<td>8.96%</td>
<td>0.0005%</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>788210</td>
<td>708515</td>
<td>708515</td>
<td>10.11%</td>
<td>0.0001%</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>799790</td>
<td>701733</td>
<td>701729</td>
<td>12.26%</td>
<td>0.0006%</td>
<td></td>
</tr>
</tbody>
</table>

**Table 2**  Different upper bounds for total expected revenue

<table>
<thead>
<tr>
<th>Case #</th>
<th>DLP REV</th>
<th>DLP16 REV</th>
<th>DCOMP REV</th>
<th>DCOMP1 REV</th>
<th>DCOMP1 bound</th>
<th>OPT-GAP</th>
<th>DCOMP1 Revenue Gains</th>
<th>%-%DLP</th>
<th>%-%DLP16</th>
<th>%-%DCOMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>334132</td>
<td>346789</td>
<td>347657</td>
<td>336655</td>
<td>355953</td>
<td>-2.68%</td>
<td>3.75%</td>
<td>-0.04%</td>
<td>-0.20%</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>306140</td>
<td>423445</td>
<td>425001</td>
<td>424262</td>
<td>438413</td>
<td>-3.34%</td>
<td>7.10%</td>
<td>0.19%</td>
<td>-0.17%</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>462636</td>
<td>493368</td>
<td>497566</td>
<td>497166</td>
<td>515024</td>
<td>-3.59%</td>
<td>7.46%</td>
<td>0.77%</td>
<td>-0.08%</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>529629</td>
<td>553200</td>
<td>558914</td>
<td>559079</td>
<td>581352</td>
<td>-3.98%</td>
<td>5.56%</td>
<td>1.06%</td>
<td>0.03%</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>563262</td>
<td>599159</td>
<td>603689</td>
<td>604945</td>
<td>626621</td>
<td>-3.58%</td>
<td>7.40%</td>
<td>0.97%</td>
<td>0.21%</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>595091</td>
<td>634115</td>
<td>637371</td>
<td>637865</td>
<td>659056</td>
<td>-3.32%</td>
<td>7.19%</td>
<td>0.59%</td>
<td>0.08%</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>627520</td>
<td>660672</td>
<td>664410</td>
<td>665188</td>
<td>686684</td>
<td>-3.23%</td>
<td>6.00%</td>
<td>0.68%</td>
<td>0.12%</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>660519</td>
<td>671735</td>
<td>682651</td>
<td>682926</td>
<td>698576</td>
<td>-2.29%</td>
<td>3.39%</td>
<td>0.83%</td>
<td>0.04%</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>672905</td>
<td>689201</td>
<td>692811</td>
<td>692894</td>
<td>708515</td>
<td>-2.25%</td>
<td>2.97%</td>
<td>0.54%</td>
<td>0.01%</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>694423</td>
<td>696927</td>
<td>697668</td>
<td>697667</td>
<td>701729</td>
<td>-0.58%</td>
<td>0.47%</td>
<td>0.11%</td>
<td>0.00%</td>
<td></td>
</tr>
</tbody>
</table>

**Table 3**  Simulated average revenues for different control policies
### Table 4  Empirical average occupancy for different control policies

<table>
<thead>
<tr>
<th>Case #</th>
<th>DLP</th>
<th>DLP16</th>
<th>DCOMP</th>
<th>DCOMP1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.9358</td>
<td>0.9794</td>
<td>0.9839</td>
<td>0.9814</td>
</tr>
<tr>
<td>2</td>
<td>0.8925</td>
<td>0.9725</td>
<td>0.9854</td>
<td>0.9828</td>
</tr>
<tr>
<td>3</td>
<td>0.8812</td>
<td>0.9622</td>
<td>0.9826</td>
<td>0.9791</td>
</tr>
<tr>
<td>4</td>
<td>0.8877</td>
<td>0.9479</td>
<td>0.9698</td>
<td>0.9666</td>
</tr>
<tr>
<td>5</td>
<td>0.8472</td>
<td>0.9228</td>
<td>0.9423</td>
<td>0.9388</td>
</tr>
<tr>
<td>6</td>
<td>0.8124</td>
<td>0.8841</td>
<td>0.9028</td>
<td>0.9016</td>
</tr>
<tr>
<td>7</td>
<td>0.7815</td>
<td>0.8415</td>
<td>0.8602</td>
<td>0.8584</td>
</tr>
<tr>
<td>8</td>
<td>0.7651</td>
<td>0.7967</td>
<td>0.8141</td>
<td>0.8135</td>
</tr>
<tr>
<td>9</td>
<td>0.7204</td>
<td>0.7533</td>
<td>0.7661</td>
<td>0.7659</td>
</tr>
<tr>
<td>10</td>
<td>0.7051</td>
<td>0.7118</td>
<td>0.7148</td>
<td>0.7149</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Case #</th>
<th>Base case</th>
<th>Scale down by 2</th>
<th>Scale down by 5</th>
<th>Scale down by 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.55%</td>
<td>0.83%</td>
<td>1.51%</td>
<td>1.61%</td>
</tr>
<tr>
<td>2</td>
<td>0.37%</td>
<td>0.51%</td>
<td>0.93%</td>
<td>1.27%</td>
</tr>
<tr>
<td>3</td>
<td>0.21%</td>
<td>0.33%</td>
<td>0.55%</td>
<td>0.90%</td>
</tr>
<tr>
<td>4</td>
<td>0.13%</td>
<td>0.20%</td>
<td>0.38%</td>
<td>0.51%</td>
</tr>
<tr>
<td>5</td>
<td>0.10%</td>
<td>0.19%</td>
<td>0.47%</td>
<td>0.88%</td>
</tr>
<tr>
<td>6</td>
<td>0.05%</td>
<td>0.12%</td>
<td>0.43%</td>
<td>1.18%</td>
</tr>
<tr>
<td>7</td>
<td>0.02%</td>
<td>0.04%</td>
<td>0.18%</td>
<td>0.65%</td>
</tr>
<tr>
<td>8</td>
<td>0.03%</td>
<td>0.07%</td>
<td>0.26%</td>
<td>0.71%</td>
</tr>
<tr>
<td>9</td>
<td>0.01%</td>
<td>0.03%</td>
<td>0.09%</td>
<td>0.41%</td>
</tr>
<tr>
<td>10</td>
<td>0.02%</td>
<td>0.06%</td>
<td>0.24%</td>
<td>0.74%</td>
</tr>
</tbody>
</table>

### Table 6  Percentage revenue gains of DCOMP1 compared with DCOMP for different scale down factors
Figure 4  Percentage revenue gains of DCOMP1 compared with DCOMP for different scale down factors

<table>
<thead>
<tr>
<th>Case #</th>
<th>Base case</th>
<th>Scale down by 2</th>
<th>Scale down by 5</th>
<th>Scale down by 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.54%</td>
<td>1.18%</td>
<td>0.62%</td>
<td>0.24%</td>
</tr>
<tr>
<td>2</td>
<td>2.32%</td>
<td>1.88%</td>
<td>1.51%</td>
<td>0.98%</td>
</tr>
<tr>
<td>3</td>
<td>2.11%</td>
<td>2.03%</td>
<td>1.77%</td>
<td>1.70%</td>
</tr>
<tr>
<td>4</td>
<td>2.01%</td>
<td>1.92%</td>
<td>1.93%</td>
<td>1.50%</td>
</tr>
<tr>
<td>5</td>
<td>1.44%</td>
<td>1.46%</td>
<td>1.44%</td>
<td>1.32%</td>
</tr>
<tr>
<td>6</td>
<td>1.06%</td>
<td>1.18%</td>
<td>1.26%</td>
<td>1.30%</td>
</tr>
<tr>
<td>7</td>
<td>0.98%</td>
<td>1.06%</td>
<td>1.19%</td>
<td>1.07%</td>
</tr>
<tr>
<td>8</td>
<td>0.80%</td>
<td>0.92%</td>
<td>0.98%</td>
<td>0.83%</td>
</tr>
<tr>
<td>9</td>
<td>0.57%</td>
<td>0.64%</td>
<td>0.68%</td>
<td>0.67%</td>
</tr>
<tr>
<td>10</td>
<td>0.23%</td>
<td>0.29%</td>
<td>0.30%</td>
<td>0.16%</td>
</tr>
</tbody>
</table>

Table 7  Percentage revenue gains of DCOMP1 compared with DLP16 for different scale down factors

Figure 5  Percentage revenue gains of DCOMP1 compared with DLP16 for different scale down factors