Assessing the Value of Dynamic Pricing in Network Revenue Management

Dan Zhang
Desautels Faculty of Management, McGill University
dan.zhang@mcgill.ca

Zhaosong Lu
Department of Mathematics, Simon Fraser University
zhaosong@sfu.ca

Dynamic pricing for a network of resources over a finite selling horizon has received considerable attention in recent years, yet few papers provide effective computational approaches to solve the problem. We consider a resource decomposition approach to solve the problem and investigate the performance of the approach in a computational study. We compare the performance of the approach to three alternative approaches: static pricing, static bid-price control, and choice-based availability control. Our numerical results show that dynamic pricing policies from the network resource decomposition can achieve significant revenue lift compared with the other three alternatives. In addition, choice-based availability control is ineffective even compared with static pricing and static bid-price control.

This version: August 4, 2009

---

1. Introduction

Dynamic pricing where product prices are changed periodically over time to maximize revenue has received considerable attention in research and application in recent years. As early as 2002, Hal Varian proclaimed that "dynamic pricing has become the rule" (Schrage 2002). However, many revenue management (RM) applications are based on product availability control where product prices are fixed and product availability is adjusted dynamically over time. Other, even simpler approaches often find wide adoption in practice, such as static pricing and bid-price controls.

There are many reasons these simpler approaches are more desirable than full-scale dynamic pricing. First of all, companies may not have full pricing power, especially when their products are not sufficiently differentiated from competitive offerings. Second, dynamic pricing faces customer acceptance issue (Phillips 2005). If not properly implemented, dynamic pricing can alienate customers because different customers can pay very different prices for essentially the same product. Finally, computing and implementing a dynamic pricing strategy can be much more complicated than these simpler alternatives. Indeed, despite years of research in dynamic pricing, practical solution approaches for dynamic pricing problem that involves multiple resource and product types are very limited (Bitran and Caldentey 2003).

Given the practical limitations of dynamic pricing, a pivotal research question is whether simpler alternatives can achieve revenue close to what can be achieved via dynamic pricing. This paper attempts to answer this question via a computational study. We compare dynamic pricing to three alternatives: static pricing, bid-price controls, and choice-based availability control.

A static pricing strategy fixes the price of each product at the beginning of the selling horizon. Static pricing strategy is clearly attractive from the implementation point of view as it does not involve periodical revision of prices. A bid-price control strategy assigns a value (bid-price) to a unit of each resource as the basis of a pricing strategy. There are many variants of the approach and we will discuss a specific approach later in the paper. Choice-based availability control is motivated by the recent literature on choice-based network revenue management. In choice-based availability control, product prices are fixed and product availability are controlled over time. An important aspect of the approach is the enriched demand model where customers are assumed to choose among all available products according to pre-specified choice probabilities. The demand model can be viewed as a generalization to the widely used independent demand model where customers belongs to different "classes" each requesting a specific product. The recent work of Liu and van Ryzin (2008) shows that choice-based availability control can significantly improve revenue compared with models based on independent demand model.

In order to achieve our research goals, it is necessary to compute a reasonable dynamic pricing policy. Note that computing an optimal dynamic pricing policy is very difficult even for relatively small problem. In fact, even for conceptually simpler models, such as network RM model with independent demand, existing research and application rely on heuristics. The widely used dynamic programming formulation for network RM suffers from the curse of dimensionality as the state space grows exponentially with the number of resources. We adopt a resource decomposition approach to decompose the network problem into a collection of single resource problems, which are then solved and subsequently used to provide approximate dynamic pricing policies. The approach uses dual values from a deterministic approximation model, which is a constrained nonlinear programming problem, for which we proposed an augmented Lagrangian approach. The approach is provably convergent and is quite efficient even for relatively large problems in our numerical experiment. We believe the overall approach has the potential to be used for realistic sized problems.

As a by-product of the decomposition approach, we show that it leads to an upper bound on revenue. In our numerical study, the bound is tighter than the upper bound from the deterministic approximation. Therefore, it provides a better benchmark in our numerical study.

A central component of our approach is the solution of the deterministic approximation model, which is a constrained nonlinear programming problem. Deterministic approximation is widely used in revenue management. Deterministic approximation of the network revenue management with independent demand leads to a linear programming formulation. Similarly, deterministic approximation of the choice-based network revenue management also leads to a linear programming formulation. Nevertheless, very few papers consider deterministic approximation which results in constrained nonlinear programming models. Consequently, algorithms for solving such nonlinear formulations are rarely discussed in the literature. Our experience suggests that nonlinear programming problems that are reasonably structured are still practical and should be considered as a serious alternative in the research and application of RM.

1.1 Literature Review

Our research is relevant to several different streams of work in the area of revenue management and pricing. A comprehensive review of the revenue management literature is given by Talluri and van Ryzin (2004b). Dynamic pricing is often considered as a sub-area of revenue management and has grown considerably in recent years. Two excellent review articles on dynamic pricing are offered by Bitran and Caldentey (2003) and Elmaghraby and Keskinocak (2003).

Early work in the area of revenue management focuses on quantity-based availability control, such as booking-limit type policies; see, for example, Belobaba (1989) and Brumelle and McGill (1993). The work assumes customers belong to different fare classes each paying a fixed fare and the decisions are the booking limits for each fare class. Even though the above cited work considers single resource problems, they can be extended to network setting via approaches such as fare proration and virtual nesting (Talluri and van Ryzin 2004b).

Dynamic pricing models differ from the quantity-based models in that they assume product prices can be adjusted within a given price set. Gallego and van Ryzin (1994) consider the dynamic pricing problem for selling a finite inventory of a given product within a finite selling horizon. Their work inspired much follow-up research for the problem (Bitran and Mondschein 1997, Zhao and Zheng 2000, Maglaras and Meissner 2006).

Relatively few papers consider dynamic pricing problem for network RM. Gallego and van Ryzin (1997) consider the dynamic pricing for network RM and establish bounds from deterministic versions of the problem and show useful heuristic approaches to the problem from the bounds. The deterministic approximation model in the current paper is conceptually the same as the one in Gallego and van Ryzin (1997) and therefore constitutes an upper bound on the optimal revenue. Their results show that static pricing can do relatively well when the problem is relatively large, which is verified by our numerical results using multinomial logit demand. Nevertheless, we show that a heuristic dynamic pricing policy can do much better, producing revenues up to 8% higher than static pricing policies. The reported revenue gap between dynamic pricing and static pricing is rather significant for most RM applications. Therefore, we argue that dynamic pricing should be considered whenever possible in practice. In an earlier paper, Dong et al. (2007) consider the dynamic pricing problem for substitutable products. The model they studied can be viewed as a dynamic pricing problem for a multi-resource single leg network. However, their focus is on structural analysis and their approach cannot be easily extended to the network setting. Zhang and Cooper (2009) consider the dynamic pricing problem for substitutable flights on a single leg. They provide bounds and heuristics for the problem. In contrast to the current paper, they assume the price for each product can be chosen from a discrete set.

Much work in the network RM literature considers availability control based RM approaches where fare prices are fixed. Classic approaches assume customers belong to different fare classes and the decisions to make are availability of different fare classes (Talluri and van Ryzin 1998, Cooper 2002). In recent years, this line of research has been expanded to consider customer choice behavior among different fare classes (Talluri and van Ryzin 2004a, Zhang and Cooper 2005, Liu and van Ryzin 2008, Zhang and Adelman 2008). Models that consider customer choice behavior can lead to much higher revenue than models based on independent demand by customer class assumption (Liu and van Ryzin 2008). However, we demonstrate numerically that the performance of choice-based RM is rather ineffective, beaten by static pricing policies in our numerical example, when the static prices are appropriately chosen in advance. This observation is consistent to the popular view that availability control is most useful when the (fixed) prices are not properly chosen (Gallego and van Ryzin 1994).

Bid-price control is widely adopted in revenue management practice. Talluri and van Ryzin (1998) establish theoretical properties for the use of such policies. One of the appeals of bid-price control lies in its simplicity relative to other control approaches. It is common to generate bid-prices from simpler approximations, notably the deterministic approximation. A popular approach to generate bid-prices in network revenue management is the deterministic linear program (Williamson 1992), where the shadow prices for the capacity constraints are taken as the bid-prices. In the choice-based revenue management, bid-prices can be generated from the choice-based linear program (Liu and van Ryzin 2008). In these papers, product prices are fixed, which is key to the linear programming formulation. In our setup, however, since product prices are decision variables, deterministic approximation of the problem is nonlinear. Nevertheless, we can use a similar idea as in these papers and take the dual variables for capacity constraints as bid-prices.

Approximate dynamic programming (Bertsekas and Tsitsiklis 1996, Powell 2007) is an active research area that has received considerable attention in recent years. A number of authors consider approximate dynamic programming approaches in network RM. Adelman (2007) considers a linear programming formulation of a finite horizon dynamic program and solves the problem under a linear functional approximation for the value function to obtain time-dependent bid-prices in the network RM context. This research is extended to the network RM with customer choice in the follow-up work by Zhang and Adelman (2008). Topaloglu (2007) considers a Lagrangian relaxation approach to dynamic programming formulation of the network RM problem. To the best of our knowledge, prior research has not applied the approach to dynamic pricing in the network setting. There are two obstacles when applying the approach to dynamic pricing problem. First, the corresponding mathematical programming formulation of the dynamic program is not a linear program as in the earlier work. Nevertheless, we show that most existing theoretical results can be carried over. Second, the deterministic formulation of the problem is a constrained nonlinear program, which requires solution techniques different from the deterministic linear programming formulation in earlier work. We show that this issue can be overcome under multinomial logit demand model with pricing, which can be transformed to a convex optimization problem for which we give an efficient solution approach.

We assume customer demand follows multinomial logit (MNL) demand function, which is a widely used demand function in the research and practice of RM (Phillips 2005). General references on MNL demand models are given by Ben-Akiva and Lerman (1985) and Anderson et al. (1992). One advantage of the MNL demand function is that it can be easily linked to the MNL choice models considered in the literature on choice-based revenue management (Talluri and van Ryzin 2004b, Liu and van Ryzin 2008, Zhang and Adelman 2008), which enables us to compare dynamic pricing to choice-based availability control.

1.2 Overview of Results and Outline

Dynamic pricing for a network of resources is an important research problem, but is notoriously difficult to solve. This paper considers a network resource decomposition approach to solve the problem. A central element of such an approach is a deterministic approximation model, which turns out to be a constrained nonlinear programming problem. We show that under a particular class of demand models, called multinomial logit demand model with disjoint consideration sets, the problem can be reduced to convex programming problem, for which we give an efficient and provably convergent solution algorithm.

We compare the dynamic pricing policy from the network resource decomposition with three alternative control approaches: static pricing, bid-price control, and choice-based availability control. The static pricing policy and the bid-price control policy come from the primal and dual solutions of the nonlinear programming problem, extending similar approaches in the literature.

The performance of the different approaches is compared by simulating the resulting policies on a set of randomly generated problem instances. Our numerical results show that the dynamic pricing policy can perform significantly better than the static pricing policy and bid-price control policy, leading to revenue improvement in the order of 3-8%. Such revenue improvement is quite significant in many revenue management context, and justifies the use of dynamic pricing policies. On the other hand, static pricing policy and bid-price policy perform better than choice-based availability control, suggesting the latter is rather ineffective when the product prices are appropriately chosen. Our results therefore emphasizes the importance of pricing decision as the main driver of superior revenue performance.

This paper makes two contributions. First, we introduce the dynamic programming decomposition approach to the dynamic pricing problem for a network of resources. Despite the popularity of dynamic pricing in research, few papers provide effective computational approaches to solve the problem. One exception is the paper by Dong et al. (2007). However, their focus is on exact analysis and the associated managerial insights. Exact analysis has very high computational cost and is not feasible for relatively large problems. Second, we perform a computational study to compare the policy performance of different approaches. The performance of full scale dynamic pricing is compared to static pricing, bid-price control, and choice-based availability control. It is established in the literature that the choice-based approach leads to significant revenue improvement compared with the independent demand model where customers are classified into classes each requesting one particular product. Performance comparison of dynamic pricing and choice-based approach with fixed prices is not available in the literature. Our results fill this gap.

The remainder of the paper is organized as follows. Section 2 introduces the model. Section 3 considers the deterministic nonlinear programming formulation and introduces a solution approach for a class of MNL demand model. Section 4 considers a dynamic programming decomposition approach. Section 5 introduces the choice-based availability control model. Section 6 reports numerical results and Section 7 summarizes.

2. Model Formulation

We consider the dynamic pricing problem on 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 notation, we use $j \in A_i$ to indicate product j uses resource i and $i \in A^j$ to indicate 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 λ and the probability of no customer arrival is $1-\lambda$. 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:

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

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 $v_{T+1}(x) = 0\ \forall x$ and $v_t(0) = 0\ \forall t$. In the above, $R_t(x) = \times_{j=1}^n R_{t,j}(x)$, where $R_{t,j}(x) = \Re_+$ if $x \geq A^j$ and $R_{t,j}(x) = \{r_\infty\}$ otherwise. The price r_∞ 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**) could be difficult to analyze mainly for two reasons: the curse of dimensionality and the complexity of the maximization in the Bellman equation. We note that (**DP**) generalizes the work of Dong et al. (2007) to the network case. Dong et al. (2007) show that intuitive structural properties does not even hold in their model where each product consumes one unit of one resource. Furthermore, 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. Deterministic Nonlinear Programming Formulation

3.1 Formulation

The use of 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 was 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 fluid with rate λ. 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 time horizon $[0, T]$. The deterministic model can be formulated as

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

In the above, (1) is a resource constraint where the inequality holds componentwise. The Lagrangian multipliers π associated with constraint (1) can be interpreted as the value of an additional unit of each resource. The solution to (**NLP**) 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 π can be used as bid-prices. Finally, as we will show later, the vector π can be used in a dynamic programming decomposition approach.

Conceptually, (**NLP**) is the same as the deterministic formulation considered in Gallego and van Ryzin (1997). They show that the solution of the problem is a bound on the optimal revenue of (**DP**). For certain special cases, for example when $P_j(r)$ is linear and the objective function of (**NLP**) is concave, the problem (**NLP**) is a convex quadratic programming problem and therefore is easy to handle. However, such a linear demand function is quite restrictive. For general demand function, it is not clear whether (**NLP**) is a convex optimization problem. In the following, we consider the solution of (**NLP**) under multinomial logit (MNL) demand model. We first introduce the MNL demand model in Section 3.2 and then give an efficient solution approach in Section 3.3.

3.2 Multinomial Logit Demand Model

Multinomial logit (MNL) demand model has been widely used in economics and marketing; see Anderson et al. (1992) for a comprehensive review. Choice models based on the MNL demand models, often called MNL choice model, have also been used extensively in the recent RM literature. Liu and van Ryzin (2008) introduce the so-called multinomial logit choice model with disjoint consideration sets. Their choice model (like all other choice models) assumes the product prices are fixed. Here we consider an extension of the model to the pricing case. Let $N = \{1, \dots, n\}$ denote the set of products. Customers are assumed to belong to L different customer segments. An arriving customer in each period belongs to segment l with probability γ_l with $\sum_{l=1}^{L} \gamma_l = 1$. Therefore, within each period, there is a segment l customer with probability $\lambda_l = \gamma_l \lambda$ with $\lambda = \sum_{l=1}^{L} \lambda_l$.

A customer in segment l considers products in the set $C_l \subseteq N$. Within each segment, the choice probability is described by an MNL model as follows. Let r denote the vector of prices for products. A segment l customer chooses product j with probability

$$P_{lj}(r) = \begin{cases} \frac{e^{(u_{lj}-r_j)/\mu_l}}{\sum_{k \in C_l} e^{(u_{lk}-r_k)/\mu_l} + e^{u_{l0}/\mu_l}}, & \text{if } j \in C_l, \\ 0, & \text{if } j \notin C_l. \end{cases}$$

In the above, the parameters μ_l, u_{lj}, and u_{l0} are constants. A segment l customer purchases nothing with probability

$$P_{l0}(r) = \frac{e^{u_{l0}/\mu_l}}{\sum_{k \in C_l} e^{(u_{lk}-r_k)/\mu_l} + e^{u_{l0}/\mu_l}}.$$

The choice model described here is called an *MNL model with disjoint consideration sets* if $C_l \cap C_{l'} = \emptyset$ for any two segments l and l'.

We assumed the seller is endowed with the value of γ but cannot distinguish customers from different segments upon arrival. It follows that

$$P_j(r) = \sum_{l=1}^{L} \gamma_l P_{lj}(r).$$

3.3 Solution to (NLP) under multinomial logit model

We next propose a suitable approach to find a solution of (**NLP**) and its vector of Lagrangian multipliers π for the MNL model with disjoint consideration sets. In view of the expressions of $P_j(r)$ and $P_{lj}(r)$, it follows that (**NLP**) is equivalent to

$$(\mathbf{NLP_r}) \qquad \max_{r \geq 0} \; d \sum_{l=1}^{L} \sum_{j \in C_l} \gamma_l r_j P_{lj}(r)$$
$$\text{s.t.} \quad \sum_{l=1}^{L} \sum_{j \in C_l} A^j P_{lj}(r) \leq c/d.$$

Hanson and Martin (1996) show that the objective function in ($\mathbf{NLP_r}$) is not quasiconcave. However, by virtue of the expression of $P_{lj}(r)$ and using a similar argument as in Dong et al. (2007), we can show that the expression

$$r_j(P) = \mu_{lj} - \mu_{l0} - \mu_l \ln P_{lj} + \mu_l \ln P_{l0} \qquad \forall j \in C_l,\ l = 1, \ldots, L,$$

gives the inverse mapping of $P_{lj}(r)$. Thus, instead of using the vector r as decision variables for ($\mathbf{NLP_r}$), we can perform the above change of variables and use the vector P as decision variables. The resulting reformulation of ($\mathbf{NLP_r}$) is given as follows

$$\begin{array}{lll} (\mathbf{NLP_p}) \qquad & \max\limits_{P \geq 0} & d \sum\limits_{l=1}^{L} \sum\limits_{j \in C_l} \gamma_l P_{lj} (\mu_{lj} - \mu_{l0} - \mu_l \ln P_{lj} + \mu_l \ln P_{l0}) \\ & \text{s.t.} & \sum\limits_{l=1}^{L} \sum\limits_{j \in C_l} A^j P_{lj} \leq c/d, \\ & & P_{l0} + \sum\limits_{j \in C_l} P_{lj} = 1, \quad l = 1, \ldots, L. \end{array}$$

Similarly as in Dong et al. (2007), we can show that ($\mathbf{NLP_p}$) is a concave maximization problem. Also, we can observe that ($\mathbf{NLP_p}$) shares with ($\mathbf{NLP}$) the same vector of Lagrangian multipliers π for the inequality constraints. We next propose an augmented Lagrangian method for finding a solution of ($\mathbf{NLP_p}$) and its vector of Lagrangian multipliers π. Before proceeding, let

$$\begin{aligned} f(P) &= \sum_{l=1}^{L} \sum_{j \in C_l} \gamma_l P_{lj} (\mu_{lj} - \mu_{l0} - \mu_l \ln P_{lj} + \mu_l \ln P_{l0}), \\ g(P) &= \sum_{l=1}^{L} \sum_{j \in C_l} A^j P_{lj} - c/d. \end{aligned}$$

Furthermore, for each $\varrho > 0$, let

$$L_\varrho(P, \pi) = f(P) + \frac{1}{2\varrho} (\|[\pi + \varrho g(P)]^+\|^2 - \|\pi\|^2).$$

In addition, define the set

$$\Delta = \{P \in \Re_+^{\tilde{n}} : P_{l0} + \sum_{j \in C_l} P_{lj} = 1,\ l = 1, \ldots, L\},$$

where $\tilde{n} = L + \sum_{l=1}^{L} |C_l|$. Also, we define the projection operator $\text{Proj}_\Delta : \Re^{\tilde{n}} \to \Delta$ as follows

$$\text{Proj}_\Delta(P) = \arg\min_{\tilde{P} \in \Delta} \|\tilde{P} - P\|, \qquad \forall P \in \Re^{\tilde{n}}.$$

We are now ready to present an augmented Lagrangian method for solving ($\mathbf{NLP_p}$).

Augmented Lagrangian method for ($\mathbf{NLP_p}$):

Let $\{\epsilon_k\}$ be a positive decreasing sequence. Let $\pi^0 \in \Re^m_+$, $\varrho_0 > 0$, and $\sigma > 1$ be given. Set $k = 0$.

1) Find an approximate solution $P^k \in \Delta$ for the subproblem

$$\min_{P \in \Delta} L_{\varrho_k}(P, \pi^k) \tag{2}$$

satisfying $\|\mathrm{Proj}_\Delta(P - \nabla_P L_{\varrho_k}(P, \pi^k)) - P\| \le \epsilon_k$.

2) Set $\pi^{k+1} := [\pi^k + \varrho_k g(x^k)]^+$ and $\varrho_{k+1} := \sigma \varrho_k$.

3) Set $k \leftarrow k+1$ and go to step 1).

end

We now state a result regarding the global convergence of the above augmented Lagrangian method for ($\mathbf{NLP_p}$). Its proof is similar to the one of Theorem 6.7 of Ruszczyński (2006).

THEOREM 1. *Assume that $\epsilon_k \to 0$. Let $\{P^k\}$ be the sequence generated by the above augmented Lagrangian method. Suppose that a subsequence $\{P^k\}_{k \in K}$ converges to P^*. Then the following statements hold:*

(a) P^* is a feasible point of ($\mathbf{NLP_p}$*);*

(b) The subsequence $\{\pi^{k+1}\}_{k \in K}$ is bounded, and each accumulation point π^* of $\{\pi^{k+1}\}_{k \in K}$ is a vector of Lagrange multipliers corresponding to the inequality constraints of ($\mathbf{NLP_p}$*).*

To make the above augmented Lagrangian method complete, we need a suitable method for solving the subproblem (2). Since the set Δ is simple enough, the spectral projected gradient (SPG) method proposed in Birgin et al. (2000) can be suitably applied to solve (2). The only nontrivial step of the SPG method for (2) lies in computing $\mathrm{Proj}_\Delta(P)$ for a given $P \in \Re^{\tilde{n}}$. In view of the definitions of Δ and $\mathrm{Proj}_\Delta(P)$ and using the fact that $C_l \cap C'_l = \emptyset$ for any two distinct segments l and l', we easily observe that $\mathrm{Proj}_\Delta(P)$ can be computed by solving L subproblems of the form

$$\min_x \left\{ \frac{1}{2}\|g - x\|^2 : \sum_i x_i = 1, x \ge 0 \right\}, \tag{3}$$

where g is a given vector. By the first-order optimality (KKT) conditions, x^* is the optimal solution of (3) if and only if there exists a scalar λ such that $\sum_i x_i^* = 1$ and x^* solves

$$\min_x \left\{ \frac{1}{2}\|g - x\|^2 - \lambda \left(\sum_i x_i - 1 \right) : x \ge 0 \right\}. \tag{4}$$

Given any λ, clearly the optimal solution of (4) is $x^*(\lambda) = \max(g + \lambda e, 0)$, where e is a all-one vector. Thus, the optimal solution x^* of (3) can be obtained by finding a root to the equation $e^T[\max(g + \lambda e, 0)] - 1 = 0$, which can be readily computed by bisection method.

4. Dynamic Programming Decomposition

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

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

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. 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). In the following, we consider a dynamic programming decomposition approach to solve the problem, which is shown to be equivalent to particular functional approximation approach.

We introduce a dynamic programming decomposition approach to solve (**DP**) based on the dual variables π in (**NLP**). For each i, t, x, $v_t(x)$ can be approximated by

$$v_t(x) \approx v_{t,i}(x_i) + \sum_{k \neq i} x_k \pi_k. \tag{5}$$

Therefore, the value $v_t(x)$ is approximated by the sum of a nonlinear term of resource i and linear terms of all other resources. Note $v_{t,i}(x_i)$ can be interpreted as the approximate value of x_i seats on resource i and $x_k \pi_k$ can be interpreted as the value of resource k. Using (5) in (**DP**) and simplifying, we obtain

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

The boundary conditions are $v_{T+1,i}(x) = 0 \; \forall x$ and $v_{t,i}(0) = 0 \; \forall t$. The set of m one dimensional dynamic programs can be solve to obtain values of $v_{t,i}(x_i)$ for each i.

Next, we show that the approximation scheme (5) yields an upper bound. We first note that (**DP**$_\mathbf{i}$) can be written as the following semi-infinite linear program:

$$(\mathbf{LP_i}) \qquad \min_{v_{t,i}(\cdot)} v_{1,i}(c_i)$$

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

PROPOSITION 2. *For each i, let $v^*_{t,i}(\cdot)$ and $\hat{v}_{t,i}(\cdot)$ be an optimal solution and a feasible solution to ($\mathbf{LP_i}$), respectively. Then*

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

Proof. It suffices to show

$$\hat{v}_{1,i}(c_i) + \sum_{k\neq i} c_k\pi_k \geq v^*_{1,i}(c_i) + \sum_{k\neq i} c_k\pi_k \geq v_1(c)$$

for each i. The first inequality above follows from the optimality of $v^*_{1,i}(c_i)$. The second inequality follows from Proposition 1 by observing that $\{v^*_{t,i}(x_i) + \sum_{k\neq i} x_k\pi_k\}_{\forall t,x}$ is feasible for ($\mathbf{LP}$). ■

PROPOSITION 3. *For each i, let $v^*_{t,i}(\cdot)$ be an optimal solution to ($\mathbf{LP_i}$) and let $v^\dagger_{t,i}(\cdot)$ be an optimal solution to ($\mathbf{DP_i}$). Then $v^*_{1,i}(c_i) = v^\dagger_{1,i}(c_i)$ for all i, t, x.*

Proof. First, it can be shown by induction that $v^*_{t,i}(x_i) \geq v^\dagger_{t,i}(x_i)$ for all t, i, x_i. Observe that $v^*_{T+1,i}(x_i) = v^\dagger_{T+1,i}(x_i) = 0$. Therefore, the inequalities hold for $T+1$. Now suppose the inequality holds for $t+1$. It follows from the constraint in ($\mathbf{LP_i}$) that

$$\begin{aligned} v^*_{t,i}(x_i) &\geq \left\{ \sum_{j=1}^{n} \lambda P_j(r_t) \left[r_{t,j} - \sum_{k\neq i} a_{kj}\pi_k + v^*_{t+1,i}(x_i - a_{ij}) - v^*_{t+1,i}(x_i) \right] \right\} + v^*_{t+1,i}(x_i) \\ &\geq \left\{ \sum_{j=1}^{n} \lambda P_j(r_t) \left[r_{t,j} - \sum_{k\neq i} a_{kj}\pi_k + v^\dagger_{t+1,i}(x_i - a_{ij}) - v^\dagger_{t+1,i}(x_i) \right] \right\} + v^\dagger_{t+1,i}(x_i). \end{aligned}$$

In the above, the second inequality follows from inductive assumption. Since the inequality holds for all $r_t \in R_t(x)$, we have

$$\begin{aligned} v^*_{t,i}(x_i) &\geq \max_{r_t \in R_t(x)} \left\{ \sum_{j=1}^{n} \lambda P_j(r_t) \left[r_{t,j} - \sum_{k\neq i} a_{kj}\pi_k + v^\dagger_{t+1,i}(x_i - a_{ij}) - v^\dagger_{t+1,i}(x_i) \right] \right\} \\ &= v^\dagger_{t,i}(x_i). \end{aligned}$$

It follows that $v^*_{1,i}(c_i) \geq v^\dagger_{1,i}(c_i)$.

On the other hand, from the optimality equations in ($\mathbf{DP_i}$) and the constraints in ($\mathbf{LP_i}$), $v^\dagger_{t,i}(\cdot)$ is feasible for ($\mathbf{LP_i}$). This implies that $v^\dagger_{1,i}(c_i) \geq v^*_{1,i}(c_i)$.

Combining the above leads to $v^\dagger_{1,i}(c_i) = v^*_{1,i}(c_i)$. This completes the proof. ■

Proposition 2 establishes that the solution to ($\mathbf{LP_i}$) provides an upper bound to the value function $v_1(c)$ of ($\mathbf{DP}$). Proposition 3 implies that it suffices to solve ($\mathbf{DP_i}$) to obtain the bound. The decomposition bound in Proposition 2 provides a useful benchmark in numerical studies.

5. Choice-based Network Revenue Management

As a benchmark, we would like to compare the performance of dynamic pricing with a choice-based availability control as considered in Liu and van Ryzin (2008). To conduct a meaningful comparison, we assume the firm first solves (**NLP**) and uses the optimal solution as the prices in the subsequent choice-based formulation. Suppose the price vector determined from (**NLP**) is denoted by the vector f.

After f is determined, a dynamic programming model can be formulated as follows; see Liu and van Ryzin (2008) and Zhang and Adelman (2008). The state at the beginning of any period t is an m-vector of unsold seats x. Let $u_t(x)$ be the maximum total expected revenue over periods $t,\ldots,T$ starting at state x at the beginning of period t. The optimality equations are

$$\begin{aligned} &(\mathbf{DP-CHOICE}) \\ u_t(x) &= \max_{S\subseteq N(x)} \left\{ \sum_{j\in S} \lambda P_j(f(S))(f_j + u_{t+1}(x-A^j)) + (\lambda P_0(f(S)) + 1 - \lambda) u_{t+1}(x) \right\} \\ &= \max_{S\subseteq N(x)} \left\{ \sum_{j\in S} \lambda P_j(f(S))[f_j - (u_{t+1}(x) - u_{t+1}(x-A^j))] \right\} + u_{t+1}(x), \qquad \forall t, x. \end{aligned}$$

The boundary conditions are $u_{T+1}(x) = 0$ for all x and $u_t(0) = 0$ for all t. In the above, the set $N(x) = \{j \in N : x \geq A^j\}$ is the set of products that can be offered when the state is x. Furthermore, $f_j(S) = f_j$ if $j \in S$ and $f_j(S) = r_\infty$ if $j \notin S$.

If (**DP-CHOICE**) is solved to optimality, the policy will perform at least as well as the static pricing policy from (**NLP**), since the latter corresponds to offer all products whenever possible in (**DP-CHOICE**). However, like (**DP**), solving (**DP-CHOICE**) is also difficult for moderate sized problem due to curse of dimensionality. Approximate solution approaches are proposed in the literature based on determined approximation. Liu and van Ryzin (2008) develop a choice-based linear programming model; see also Gallego et al. (2004). Let S denote the firm's offer set. Customer demand (viewed as continuous quantity) flows in at rate λ. If the set S is offered, product j is sold at rate $\lambda P_j(f(S))$ (i.e., a proportion $P_j(f(S))$ of the demand is satisfied by product j). Let $R(S)$ denote the revenue from one unit of customer demand when the set S is offered. Then

$$R(S) = \sum_{j\in S} f_j P_j(f(S)).$$

Note that $R(S)$ is a scalar. Similarly, let $Q_i(S)$ denote the resource consumption rate on resource i, $i = 1,\ldots,m$, given that the set S is offered. Let $Q(S) = (Q_1(S),\ldots,Q_m(S))^T$. The vector $Q(S)$ satisfies $Q(S) = AP(f(S))$, where $P(f(S)) = (P_1(f(S)),\ldots,P_n(f(S)))^T$ is the vector of purchase probabilities.

Let $h(S)$ be the total time the set S is offered. Since the demand is deterministic as seen by the model and the choice probabilities are time-homogeneous, only the total time a set is offered matters. The objective is to find the total time $h(S)$ each set S should be offered to maximize the firm's revenue. The linear program can be written as follows

$$
\begin{aligned}
(\mathbf{CDLP}) \qquad\qquad z_{CDLP} = \max_h & \sum_{S \subseteq N} \lambda R(S) h(S) \\
& \sum_{S \subseteq N} \lambda Q(S) h(S) \leq c \qquad (6) \\
& \sum_{S \subseteq N} h(S) = T \qquad (7) \\
& h(S) \geq 0, \quad \forall S \subseteq N.
\end{aligned}
$$

Note that $\emptyset \subseteq N$ so that the decision variable $h(\emptyset)$ corresponds to the total time that no products are offered. Liu and van Ryzin (2008) show that (**CDLP**) can be solved via a column generation approach for the MNL choice model with disjoint consideration sets. The dual variables associated with the resource constraint (6) can be used in dynamic programming decomposition approaches similar to the ones developed in Section 4.

6. Numerical Study

The purpose of the numerical study is two-fold. First, we would like to study the performance of the decomposition approach to dynamic pricing in the network setting. Second, perhaps more importantly, we would like to compare the performance of dynamic pricing policies to other alternative control strategies.

6.1 Problem Instances

We conduct numerical experiments using randomly generated problem instances. We consider hub-and-spoke network problems with one hub and several non-hub locations. Hub-and-spoke network is a widely used network structure in the airline industry, as it allows an airline to serve many different locations with relatively few scheduled flights through customer connections at the hub. Figure 7 shows a hub-and-spoke network with one hub and 4 non-hub locations. There are one flight scheduled from each of the two non-hub locations on the left to the hub and one flight from the hub to each of the two non-hub locations on the right. Customers can travel in the local markets from non-hub locations (on the left) to the hub or from the hub to non-hub locations (on the right). These itineraries are called local itineraries. In addition, customers can travel from the non-hub locations on the left to the non-hub locations on the right via the hub. These itineraries are called through itineraries.

We randomly generate two sets of hub-and-spoke network instances with 4 non-hub locations. In the first set of examples, which we call **HS1**, the total number of periods $T = 500$. There are 4 scheduled flights each with capacity 30, two of which are to the hub and the other two are from the hub, as shown in Figure 7. In total, there are 4 local itineraries and 4 through itineraries. Note that only one product is offered for each itinerary, and therefore there are 8 different consideration sets each with one product. The MNL choice parameters are generated as follows. The u_{lj} values for local products are generated from a uniform $[10, 100]$ distribution. The u_{lj} values for through products are given by 0.95 times the sum of the u_{lj} values on the corresponding local products. The value μ_0 is generated from uniform $[0, 20]$ distribution and μ_l is generated from uniform $[0, 100]$ distribution. The arrival probability λ is taken to be 1 in each period. The probability that an arriving customer belongs to segment l is given by $\gamma_l = X_l / \sum_{k=1}^{L} X_k$ where X_k's are independent uniform $[0, 1]$ random variables. Note that, once generated, γ_l is held constant throughout the booking horizon. This procedure is used to generate 10 different problem instances which we label case 1 through case 10.

Another set of hub-and-spoke example, which we call **HS2**, has the same network structure as **HS1**. The problem data is generated in a similar fashion except that there are two products offered for each itinerary, belonging to two different consideration sets. The u_{lj} values for the first and the second consideration sets on each itinerary are generated from uniform $[10, 1000]$ and uniform $[10, 100]$ distribution, respectively. The values of μ_l and μ_{l0} are generated from uniform $[0, 100]$ distributions. All other parameters are generated in the same way as for **HS1**. Ten different problem instances labeled case 1 through case 10 are generated.

6.2 Policies and Simulation Approach

The following policies are considered in our numerical study:

- **DCOMP**: This policy implements the dynamic programming decomposition introduced in Section 4. 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

$$v_t(x) \approx \sum_{i=1}^{m} v_{t,i}(x_i). \tag{8}$$

Using (8), we have

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

An approximate policy to (**DP**) is given by

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

- **STATIC**: This policy implements the optimal prices from (**NLP**). The product prices are fixed throughout the booking horizon; a product is not offered when demand for the product cannot be satisfied with the remaining capacity. This policy is called **STATIC** to reflect the fact that prices and availability are not changed over time (except when it is not feasible to offer a given product).
- **CDLP**: This policy implements the shadow prices of the capacity constraint in (**CDLP**) as static bid-prices. Given a bid-price vector π, a pricing policy is given by

$$r_t^* = \arg \max_{r_t \in R_t(x)} \left\{ \sum_{j=1}^{n} \lambda P_j(r_t) \left[r_{t,j} - \sum_{i=1}^{m} a_{ij} \pi_i \right] \right\}.$$

Note that even though the bid-prices are fixed, it is possible that a product is not offered even if it is feasible to satisfy demand from that product. It is also possible to implement the Lagrangian multipliers for the capacity constraint in (**NLP**) in a similar fashion, which we did in our numerical study. The performance of that policy is very close to this policy. For this reason, we only report results from **CDLP** bid prices.
- **CHOICE**: This policy implements the dynamic programming decomposition introduced in Liu and van Ryzin (2008). It is a dynamic availability control policy.

We use simulation to evaluate the performance of the different approaches. For each set of instances, we randomly generated 5000 streams of demand arrivals, where the arrival in each period can be represented by a uniform $[0, 1]$ random variable X. Given product prices r, an incoming customers chooses product j if $\sum_{k=1}^{j-1} P_k(r) \leq X < \sum_{k=1}^{j} P_k(r)$. We choose not to report information on simulation errors, which when measured by the 95% half width divided by the mean is less than 0.5% for all averages we reported below. Note that the bounds reported are exact and are not subject to simulation errors.

6.3 Results for HS1

Table 1 reports the simulated average revenues for the four policies we consider. The performance of **DCOMP** is compared against the decomposition bound. The optimality gap is the percentage gap between the **DCOMP** REV and the decomposition bound. This gap is 3-5%. It should be pointed out that the decomposition bound is tighter than the bound from **NLP**, which is reported in Table 2. We also compare the performance of **DCOMP** to **STATIC**, **CDLP**, and **CHOICE**. Note that the relative performance reported is compared against the decomposition bound; i.e., the gap is computed as the difference in average revenues divided by the decomposition bound. Observe that **CDLP** and **CHOICE** are not performing as well as **STATIC** in almost all problem instances.

This is quite surprising given that **CDLP** and **CHOICE** are dynamic capacity dependent policies, while **STATIC** (as its name suggests) is purely static. In particular, this shows that choice-based RM strategies, while effective when prices are not chosen appropriately, are not very effective when prices are optimized. On the other hand, all three alternative policies are not performing as well as the dynamic pricing strategy **DCOMP**. Indeed, **DCOMP** shows a consistent 3-5% revenue improvement across the board. In most RM setting, this improvement is quite significant. This shows that dynamic pricing strategy should be considered when possible in practice. When dynamic pricing strategy is not feasible, **STATIC** provides a very strong heuristic, considering its strong performance and its static nature.

Table 2 reports the bounds from the decomposition and **NLP**. Note that the objective of **CDLP** is the same as **NLP** when the set of optimal static prices from **NLP** are used as input to **CDLP**. Therefore, it is not necessary to report the values from **CDLP**. The decomposition problem for each leg also provides a bound, which is tighter than the bound from **NLP**. These bounds from different legs are pretty close. The decomposition bound (taken as the minimum of all bounds from each leg) is 1-3% tighter than the **NLP** bound.

6.4 Results for HS2

Table 3 reports the results for **HS2**. The general observations are in line with those from **HS1**. The sub-optimality gap of **DCOMP** is slightly smaller at 1-4%. The policy **STATIC** performs better than **CDLP** and **CHOICE** in the majority of problem instances, confirming the robustness of the policy observed in **HS1**. Dynamic pricing policy **DCOMP** shows significant revenue improvement at 3-8% against the three alternative policies, which is even more significant than what we observed in **HS1**. Note that **HS1** and **HS2** differ in that **HS2** has two customer segments with very different price sensitivity. This shows that the ability to adjust prices dynamically is more important when customers differ more significantly in their price sensitivity. Table 4 reports the bound from the dynamic programming decomposition and **NLP**.

6.5 Computational Time and Bound Performance

In this section, we report the computational time on randomly generated hub-and-spoke network instances. The network structure is similar to the one presented in Figure 7. The instances are generated in the same way as **HS1** except for different number of non-hub locations. The number of periods is in the set $\{100, 200, 400, 800\}$. The capacity is varied so that the capacity/demand ratio is approximately the same. The largest problem instance we consider has 16 non-hub locations and 80 products. Table 5 reports the CPU seconds for the different problem instances when solving

DCOMP. For the largest problem instance, the solution time is about 695 seconds (less than 12 minutes), which is still practical for real applications. Even though we do not report the computational time for **CHOICE**, we observed in our numerical experiments that its computational time is only slightly shorter. The reason is that the optimization for each dynamic programming recursion in **DCOMP** can be done very efficiently using a line search. Furthermore, (**NLP**) can be solved quickly, and the solution algorithm scales very well with time and capacity as it is a continuous optimization problem.

Table 6 reports the bounds from **NLP** and **DCOMP**. We observe that the **DCOMP** bound is always tighter than the **NLP** bound. Furthermore, the difference between the bounds tend to be larger for smaller problem instances. In the set of examples we consider, the relative difference of the two bounds can be as high as 12%. For large problem instances, the relative difference between the two bounds are relatively small. Nevertheless, as shown in our simulation results in previous subsections, the performance lift from **DCOMP** can be quite significant even when the bounds are close.

7. Summary and Future Directions

This paper studies the value of dynamic pricing by comparing with several other reasonable RM approaches, including static pricing, static bid-prices, and choice-based availability control. Our results show that dynamic pricing can lead to significant revenue lift, in the order of 3-8% in our numerical study across the board. On the other hand, choice-based availability control does not perform well compared even with static pricing. Therefore, dynamic pricing approaches should be implemented whenever possible in practice.

Our research suffers from the following limitation. First of all, the static prices considered in this research is generated from a deterministic approximation which ignores demand uncertainty. Because of this, the gap reported in this paper between dynamic pricing and static pricing may be an overestimate of the true gap between the two. In the same vein, the fixed prices feeded to the choice-based availability control is sub-optimal. Future research will benefit from more realistic modeling of static pricing.

References

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

Anderson, S. P., A. Palma, J.-F. Thisse. 1992. *Discrete Choice Theory of Product Differentiation*. The MIT Press.

Belobaba, Peter P. 1989. Application of a probabilistic decision model to airline seat inventory control. *Operations Research* **37**(2) 183–197.

Ben-Akiva, M., S. Lerman. 1985. *Discrete Choice Analysis*. MIT Press.

Bertsekas, D. P., J. N. Tsitsiklis. 1996. *Neuro-Dynamic Programming*. Athena Scientific, Belmont, MA.

Birgin, E., J. Martínez, M. Raydon. 2000. Nonmonotone spectral projected gradient methods on convex sets. *SIAM Journal on Optimization* **10**(4) 1196–1211.

Bitran, G., R. Caldentey. 2003. An overview of pricing models for revenue management. *Manufacturing and Service Operations Management* **5**(3) 203–229.

Bitran, G. R., S. V. Mondschein. 1997. Periodic pricing of seasonal products in retailing. *Management Science* **43**(1) 64–79.

Brumelle, S. L., J. I. McGill. 1993. Airline seat allocation with multiple nested fare classes. *Operations Research* **41**(1) 127–137.

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

Dong, L., P. Kouvelis, Z. Tian. 2007. Dynamic pricing and inventory control of substitute products. To appear in *Manufacturing and Service Operations Management.*

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.

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** 999–1020.

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

Hanson, Ward, Kipp Martin. 1996. Optimizing multinomial logit profit functions. *Management Science* **42**(7) 992–1003.

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.

Maglaras, C., J. Meissner. 2006. Dynamic pricing strategies for multi-product revenue management problems. *Manufacturing & Service Operations Management* **8** 136–148.

Phillips, R. 2005. *Pricing and Revenue Optimization*. Stanford University Press.

Powell, W. 2007. *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. Wiley-Interscience.

Ruszczyński, A. 2006. *Nonlinear Optimization*. Princeton University Press, Princeton and Oxford.

Schrage, M. 2002. To Hal Varian, the price is always right. *Strategy and Business* **First Quarter** http://www.strategy–business.com/press/16635507/10326.

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. 2004a. Revenue management under a general discrete choice model of consumer behavior. *Management Science* **50**(1) 15–33.

Talluri, K., G. J. van Ryzin. 2004b. *The Theory and Practice of Revenue Management*. Kluwer Academic Publishers.

Topaloglu, H. 2007. Using lagrangian relaxation to compute capacity-dependent bid prices in network revenue management. Forthcoming in *Operations Research*.

Williamson, Elizabeth L. 1992. Airline network seat control. Ph.D. thesis, Massachusetts Institute of Technology.

Zhang, D., D. Adelman. 2008. An approximate dynamic programming approach to network revenue management with customer choice. Forthcoming in *Transportation Science*.

Zhang, D., W. L. Cooper. 2005. Revenue management for parallel flights with customer-choice behavior. *Operations Research* **53** 415–431.

Zhang, D., W. L. Cooper. 2009. Pricing substitutable flights in airline revenue management. *European Journal of Operational Research* **197**(3) 848–861.

Zhao, W., Y.-S. Zheng. 2000. Optimal dynamic pricing for perishable assets with nonhomogeneous demand. *Management Science* **46**(3) 375–388.

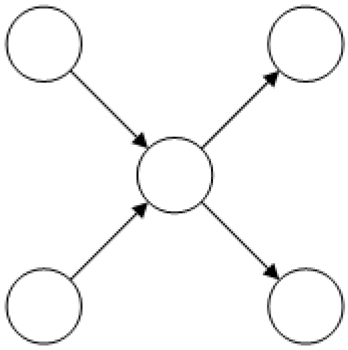

Figure 1 Hub-and-spoke network with 4 locations.

Case number	STATIC REV	CDLP REV	CHOICE REV	DCOMP REV	DCOMP Bound	OPT-GAP	DCOMP Revenue Gains		
							%STATIC	%CDLP	%CHOICE
1	12134.06	12134.06	12098.80	12588.64	13019.24	-3.31%	3.49%	3.49%	3.76%
2	11520.15	11520.15	11406.38	12077.47	12399.02	-2.59%	4.49%	4.49%	5.41%
3	12399.84	12399.84	12360.35	12864.42	13334.24	-3.52%	3.48%	3.48%	3.78%
4	14283.76	14283.76	14057.01	14827.17	15394.07	-3.68%	3.53%	3.53%	5.00%
5	11929.15	11929.15	11867.28	12509.89	12922.09	-3.19%	4.49%	4.49%	4.97%
6	10991.62	10991.62	10900.67	11455.32	11867.67	-3.47%	3.91%	3.91%	4.67%
7	13850.19	13850.19	13869.48	14495.97	15105.57	-4.04%	4.28%	4.28%	4.15%
8	11482.86	11482.86	11374.96	11961.13	12518.69	-4.45%	3.82%	3.82%	4.68%
9	12813.76	12813.76	12710.78	13263.23	13864.65	-4.34%	3.24%	3.24%	3.98%
10	9428.38	9428.38	9443.89	9760.14	10071.64	-3.09%	3.29%	3.29%	3.14%

Table 1 Simulation results for **HS1**.

Case number	$v^*_{1,i}(c_i)+\sum_{k\neq i}\pi^*_k c_k$				DCOMP bound	NLP bound	$\sum_{i=1}^n v_{1,i}(c_i)$
	$i=1$	$i=2$	$i=3$	$i=4$			
1	13077.41	13226.46	13064.35	13019.24	13019.24	13234.77	33551.11
2	12511.50	12538.33	12451.61	12399.02	12399.02	12576.09	25235.54
3	13532.94	13473.13	13334.24	13430.58	13334.24	13586.20	36922.11
4	15413.34	15394.07	15491.10	15627.06	15394.07	15654.93	37143.67
5	13050.99	12969.11	12922.09	13034.82	12922.09	13127.19	27834.69
6	11947.00	11972.10	11910.86	11867.67	11867.67	12055.79	26221.60
7	15128.17	15199.28	15237.93	15105.57	15105.57	15354.98	36053.80
8	12590.75	12518.69	12532.85	12527.12	12518.69	12707.03	27949.53
9	13864.65	13927.97	13925.02	13935.75	13864.65	14106.07	33895.00
10	10298.03	10158.30	10071.64	10182.48	10071.64	10298.27	24793.82

Table 2 Bounds and approximations for **HS1**.

Case number	STATIC REV	CDLP REV	CHOICE REV	DCOMP REV	DCOMP Bound	OPT-GAP	DCOMP Revenue Gains		
							%STATIC	%CDLP	%CHOICE
1	14358.15	14357.45	14336.73	15076.63	15570.60	-3.17%	4.61%	4.62%	4.75%
2	20096.58	20096.86	19975.05	21570.43	22038.00	-2.12%	6.69%	6.69%	7.24%
3	30732.32	30731.48	30464.40	31820.05	33033.45	-3.67%	3.29%	3.30%	4.10%
4	18064.12	18064.21	18081.95	18977.13	19647.95	-3.41%	4.65%	4.65%	4.56%
5	21820.12	21819.46	21616.20	22927.02	23797.40	-3.66%	4.65%	4.65%	5.51%
6	37033.87	37032.94	36703.77	39395.05	40565.83	-2.89%	5.82%	5.82%	6.63%
7	17298.20	17298.80	17165.57	18408.51	18946.84	-2.84%	5.86%	5.86%	6.56%
8	21315.99	21348.69	21190.00	22270.42	23157.51	-3.83%	4.12%	3.98%	4.67%
9	26783.86	26784.74	26683.89	28459.57	29321.11	-2.94%	5.72%	5.71%	6.06%
10	30433.41	30433.19	30176.23	32467.25	33115.89	-1.96%	6.14%	6.14%	6.92%

Table 3 Simulation results for **HS2**.

Case number	$v^*_{1,i}(c_i)+\sum_{k\neq i}\pi^*_k c_k$				DCOMP bound	NLP bound	$\sum_{i=1}^n v_{1,i}(c_i)$
	$i=1$	$i=2$	$i=3$	$i=4$			
1	15618.87	15570.60	15793.81	15691.80	15570.60	15853.44	35912.33
2	22038.00	22133.22	22178.58	22091.63	22038.00	22281.68	36823.64
3	33635.17	33111.00	33598.89	33033.45	33033.45	33793.69	67207.48
4	19647.95	19673.29	19819.01	19672.74	19647.95	19931.89	41731.27
5	24016.56	23824.95	23864.65	23797.40	23797.40	24145.96	42468.18
6	40770.59	40565.83	40790.28	40646.65	40565.83	40986.16	57208.16
7	19009.15	19058.86	19103.24	18946.84	18946.84	19210.81	36743.47
8	23180.02	23388.13	23418.89	23157.51	23157.51	23578.96	42348.95
9	29321.11	29412.18	29471.92	29418.04	29321.11	29707.53	49175.19
10	33322.28	33245.14	33115.89	33344.96	33115.89	33456.72	47370.80

Table 4 Bounds and approximations for **HS2**.

τ	# non-hub locations, # resources, # products							
	2,2,3		4,4,8		8,8,24		16,16,80	
	capacity per leg	**DCOMP** seconds	capacity per leg	**DCOMP** seconds	capacity per leg	**DCOMP** seconds	capacity per leg	**DCOMP** seconds
100	10	0.63	5	1.22	2	2.70	1	8.36
200	20	2.25	10	4.92	5	13.61	2	34.33
400	40	9.39	20	19.94	10	54.73	5	168.72
800	80	35.84	40	80.13	20	223.16	10	694.81

Table 5 CPU seconds for **DCOMP** in hub-and-spoke test cases.

τ	# non-hub locations, # resources, # products											
	2,2,3			4,4,8			8,8,24			16,16,80		
	capacity per leg	**NLP** bound	**DCOMP** bound	capacity per leg	**NLP** bound	**DCOMP** bound	capacity per leg	**NLP** bound	**DCOMP** bound	capacity per leg	**NLP** bound	**DCOMP** bound
100	10	2733.67	2563.27	5	1795.93	1683.13	2	1607.55	1443.85	1	1574.75	1407.46
200	20	3376.76	3250.25	10	4278.50	4123.21	5	4612.23	4420.28	2	3909.14	3718.11
400	40	7424.25	7274.92	20	6957.79	6795.66	10	9856.64	9594.96	5	9128.55	8920.96
800	80	14206.94	14069.72	40	17638.88	17403.02	20	16722.95	16499.12	10	17841.60	17592.12

Table 6 Bounds from **NLP** and **DCOMP** in hub-and-spoke test cases.