Assessing the Value of Dynamic Pricing in Network Revenue Management

Dan Zhang

Leeds School of Business, University of Colorado at Boulder, UCB 419, Boulder, Colorado 80309, dan.zhang@colorado.edu

Zhaosong Lu

Department of Mathematics, Simon Fraser University, Burnaby, BC, V5A 1S6, Canada, 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 static pricing and choice-based availability control. Our numerical results show that dynamic pricing policies from network resource decomposition can achieve significant revenue lift compared with choice-based availability control and static pricing, even when the latter is frequently resolved. As a by-product of our approach, network decomposition provides an upper bound in revenue, which is provably tighter than the well-known upper bound from a deterministic approximation.

---

1. Introduction

Dynamic pricing, whereby 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, in which product prices are fixed and product availability is adjusted dynamically over time. Static pricing, whereby the price for each product is fixed, is also frequently observed in practice.

There are many reasons for why 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 issues (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 problems that involve 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 static pricing and choice-based availability control. We also consider a version of static pricing control, which is updated periodically over the selling horizon.

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. Choice-based availability control is motivated by the recent literature on choice-based network revenue management (Talluri and van Ryzin 2004a, Zhang and Cooper 2005, Liu and van Ryzin 2008). In choice-based availability control, product prices are fixed, and product availability is controlled over time. An important aspect of the approach is the enriched demand model, whereby 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. Unlike a choice-based demand model, an independent demand model assumes demand for each product comes from different customers and the demand for a product is lost when the product is not available. The recent work of Liu and van Ryzin (2008) shows that choice-based availability control can significantly improve revenue, compared with models based on the independent demand model.

In order to achieve our research goals, it is necessary to compute a reasonable dynamic pricing policy, which is quite difficult, even for relatively small problems. In fact, even for conceptually simpler models, such as the 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 are subsequently used to provide approximate dynamic pricing policies. This 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. Since the computational time for the approach is approximately increasing linearly in the number of resources, 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, which is tighter than the upper bound from the deterministic approximation. This new upper bound 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 for network revenue management problems is widely used in research and practice of revenue management, and can at least be traced back to the earlier work of Gallego and van Ryzin (1997). Deterministic approximation of network RM with independent demand leads to a linear programming formulation. Similarly, deterministic approximation of choice-based network revenue management also leads to a linear programming formulation. The recent work of Karaesman and van Ryzin (2004) considers a network capacity allocation model with overbooking. They consider an augmented Lagrangian approach to solve a large nonlinear program, which is related to our solution approach for deterministic approximation. Our experience in this paper suggests that reasonably structured nonlinear programming problems 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 that customers belong to different fare classes, with 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 settings via approaches such as fare proration and virtual nesting (Talluri and van Ryzin 2004b).

Dynamic pricing models differ from 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 the dynamic pricing problem for network RM. Gallego and van Ryzin (1997) consider dynamic pricing for network RM and establish bounds from deterministic versions of the problem showing 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 model. Nevertheless, we show that a heuristic dynamic pricing policy can do much better, producing revenues up to 6% higher than static pricing policies. The reported revenue gap between dynamic 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 network with multiple flights sharing the same origin and destination. However, their focus is on structural analysis, and their approach cannot be easily extended to the network setting. Another related paper is the earlier work by Aydin and Ryan (2002), in which they consider a product line selection and pricing problem under the multinomial logit choice model. Zhang and Cooper (2009) consider the dynamic pricing problem for substitutable flights on a single leg. They provide bounds and heuristics for the problem.

Much work in the network RM literature considers availability control based RM ap-

proaches where fares are fixed. Classic approaches assume that customers belong to different fare classes and the decisions to make concern the 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 2009). Models that consider customer choice behavior can lead to much higher revenue than models based on independent demand by customer class assumptions (Liu and van Ryzin 2008). However, we demonstrate numerically that 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 with the popular view that availability control is most useful when (fixed) prices are not properly chosen (Gallego and van Ryzin 1994).

Bid-price control is widely adopted in revenue management practice, where a marginal value (bid-price) is assigned to each resource and a product is made available when the revenue from the product exceeds the sum of bid-prices of all resources consumed. 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), whereby shadow prices for capacity constraints are taken as the bid-prices. In choice-based revenue management, bid-prices can be generated from the choice-based linear program (Liu and van Ryzin 2008), and a bid-price control policy generates revenue maximizing product offer sets based on the bid-prices. 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 have considered 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 (2009). Topaloglu (2009) considers a Lagrangian relaxation approach to a dynamic programming formulation of the network RM problem. More recently, Erdelyi and Topaloglu (2010) apply similar techniques to dynamic pricing problems in the network setting. In their setting, the price for each product is chosen from a discrete set and the demand for each product depends on the price of the product only. As a result, the deterministic problem can be formulated as a linear program. Our work differs from theirs in the following way. First, the corresponding mathematical programming formulation of the dynamic program is not a linear program. Nevertheless, we show that most existing theoretical results can be carried over. Second, the deterministic formulation of the problem is a constrained nonlinear program, requiring solution techniques different from the deterministic linear programming formulation in their work. We show that this issue can be overcome under the multinomial logit demand model with disjoint consideration sets, which can be transformed into a convex optimization problem for which we give an efficient solution approach.

We assume that customer demand follows a 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). The specific model we use, the multinomial logit demand model with disjoint consideration sets is first introduced in Liu and van Ryzin (2008). 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 2009), 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 a multinomial logit demand model with disjoint consideration sets, the problem can be reduced to a 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, static pricing with resolving, and choice-based availability control. The static pricing policy comes from 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, leading to revenue improvement in the order of 1-6%. Such revenue improvement is quite significant in many revenue management contexts, and justifies the use of dynamic pricing policies. On the other hand, static pricing policy performs better than the choice-based availability control, suggesting that the latter is rather ineffective when the product prices are appropriately chosen. Our results, therefore, emphasize 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. As a by-product of our analysis, we show that dynamic programming decomposition leads to an upper bound on revenue, which is provably tighter than the upper bound from a deterministic approximation. 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 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, with each requesting one particular product. Performance comparison of dynamic pricing and the 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 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 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:

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

$$= \max_{r_t \in R_t(x)} \left\{ \sum_{j=1}^{n} \lambda P_j(r_t)[r_{t,j} + v_{t+1}(x - A^j) - v_{t+1}(x)] \right\} + v_{t+1}(x)$$

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

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 do 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 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 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 expected total customer arrivals over the time horizon $[0, T]$. The deterministic model can be formulated as

$$(\textbf{NLP}) \qquad \max_{r \geq 0} \quad d \sum_{j=1}^{n} r_j P_j(r)$$

$$\text{s.t.} \quad dAP(r) \leq c. \qquad (1)$$

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. 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. In the following, we consider the solution of (**NLP**) under a special multinomial logit (MNL) choice model, which is called multinomial logit choice model with disjoint consideration sets (MNLD) (Liu and van Ryzin 2008). We first introduce the demand model in Section 3.2 and then give an efficient solution approach in Section 3.3.

3.2. MNLD

The multinomial logit (MNL) choice 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 model, often called MNL choice models, have also been used extensively in the recent RM literature. The MNLD model is first introduced in Liu and van Ryzin (2008). Their choice model (like all other choice models) assumes that product prices are fixed. Here we consider an extension of the model to the pricing case. Let $N = \{1, \ldots, n\}$ 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 that 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 the multinomial logit model

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

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

Hanson and Martin (1996) show that the objective function in (**NLP$_\mathbf{r}$**) is not quasi-concave. By definition of $P_{lj}(r)$, we have that for $j \in C_l$

$$\frac{P_{lj}(r)}{P_{l0}(r)} = e^{(u_{lj}-r_j-u_{l0})/\mu_l}.$$

It follows that r can be written as functions of purchase probabilities P where

$$r_j(P) = u_{lj} - u_{l0} - \mu_l \ln P_{lj} + \mu_l \ln P_{l0}, \quad \forall j \in C_l, \ l = 1, \dots, L.$$

Note that a similar argument is used in the earlier work by Dong et al. (2007). Thus, instead of using the vector r as decision variables for (**NLP$_\mathbf{r}$**), we can perform the above change of variables and use the vector P as decision variables. The resulting reformulation of (**NLP$_\mathbf{r}$**)

is given as follows

$$(\mathbf{NLP_p}) \qquad \max_{P \geq 0} \quad d \sum_{l=1}^{L} \sum_{j \in C_l} \gamma_l P_{lj} (\mu_{lj} - \mu_{l0} - \mu_l \ln P_{lj} + \mu_l \ln P_{l0})$$

$$\text{s.t.} \quad \sum_{l=1}^{L} \sum_{j \in C_l} A^j \gamma_l P_{lj} \leq c/d,$$

$$P_{l0} + \sum_{j \in C_l} P_{lj} = 1, \quad l = 1, \ldots, L.$$

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

$$f(P) = \sum_{l=1}^{L} \sum_{j \in C_l} \gamma_l P_{lj} (\mu_{lj} - \mu_{l0} - \mu_l \ln P_{lj} + \mu_l \ln P_{l0}),$$

$$g(P) = \sum_{l=1}^{L} \sum_{j \in C_l} A^j \gamma_l P_{lj} - c/d.$$

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

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

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

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 ($\mathbf{NLP_p}$).

Note that Theorem 1 assumes a subsequence of $\{P^k\}$ converges, which can be guaranteed when the sequence $\{P^k\}$ is bounded. 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 $\text{Proj}_\Delta(P)$ for a given $P \in \Re^{\tilde{n}}$. In view of the definitions of Δ and $\text{Proj}_\Delta(P)$ and using the fact that $C_l \cap C'_l = \emptyset$ for any two distinct segments l and l', we easily observe that $\text{Proj}_\Delta(P)$ can be computed by solving L subproblems of the form

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

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

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

Given any λ, clearly the optimal solution of (4) is $x^*(\lambda) = \max(g + \lambda e, 0)$, where e is an 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 the 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:

(**LP**) $\quad \min\limits_{v_t(\cdot)} v_1(c)$

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

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; see, Adelman (2007). 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 a particular functional approximation approach.

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

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

for each t and x. 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 enlarging the feasible set $R_t(x)$ to $R_t(x_i, c_{-i})$, we obtain

$$(\mathbf{DP_i}) \quad v_{t,i}(x_i) \tag{6}$$

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

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

The boundary conditions are $v_{T+1,i}(x) = 0 \ \forall x$ and $v_{t,i}(0) = 0 \ \forall t$. 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.

Using similar techniques, the maximization in (**DP_i**) for each state x_i and time t can be reformulated as a convex optimization problem similar to (**NLP_p**), but without capacity constraints. The SPG algorithm discussed at the end of Section 3.3 can be readily applied to efficiently solve this problem. Note that the SPG algorithm is very efficient because it is a gradient projection method whose subproblem can be easily solved.

Next, we show that the approximation scheme (5) yields an upper bound. We first note that (**DP_i**) can be written as the following semi-infinite linear program:

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

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

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 (**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 (**LP**). Note that here we use the fact that $R(x) \subseteq R(x_i, c_{-i})$. ∎

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

Proof. First, it can be shown by induction that $v_{t,i}^*(x_i) \geq v_{t,i}^\dagger(x_i)$ for all t, i, x_i. Observe that $v_{T+1,i}^*(x_i) = v_{T+1,i}^\dagger(x_i) = 0$. 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

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

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

In the above, the second inequality follows from inductive assumption. Since the inequality holds for all $r_t \in R_t(x_i, c_{-i})$, we have

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

$$
= v_{t,i}^\dagger(x_i).
$$

It follows that $v_{1,i}^*(c_i) \geq v_{1,i}^\dagger(c_i)$.

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

Combining the above leads to $v_{1,i}^\dagger(c_i) = v_{1,i}^*(c_i)$. 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. Next we show that for MNLD demand, the decomposition bound is tighter than the upper bound from deterministic approximation, and therefore provides a useful benchmark in numerical studies.

Proposition 4 *For MNLD demand model, dynamic programming decomposition leads to a tighter upper bound than the deterministic approximation. That is, suppose $\{v_{t,i}^*(\cdot)\}_{\forall t,i}$ is an optimal solution to* $(\mathbf{LP_i})$, *then $v_{1,i}^*(c_i) + \sum_{k \neq i} c_k \pi_k \leq z_{NLP}$ for each i.*

Proof. First note that (**NLP**) can be transformed into a convex program with linear constraints for MNLD demand model, therefore strong duality holds; that is

$$z_{NLP} = \max_{r \geq 0} \left\{ d \sum_{j=1}^{n} r_j P_j(r) + \sum_{k=1}^{m} \pi_k \left(c_k - d \sum_{j=1}^{n} a_{kj} P_j(r) \right) \right\}$$
$$= \max_{r \geq 0} \left\{ d \sum_{j=1}^{n} P_j(r) \left(r_j - \sum_{k=1}^{m} a_{kj} \pi_k \right) + \sum_{k=1}^{m} \pi_k c_k \right\}.$$

In the above, $d = \lambda T$ is the the expected total demand.

On the other hand, it can be checked that

$$v_{t,i}(x_i) = \pi_i x_i + \sum_{\tau=t}^{T} \max_{r \geq 0} \sum_{j=1}^{n} \lambda P_j(r) \left(r_j - \sum_{k=1}^{m} a_{kj} \pi_k \right), \quad \forall t, x_i,$$

is a feasible solution to (**LP$_i$**). By Proposition 1, an upper bound of $v_{t,i}^*(c_i)$ is given by

$$v_{1,i}(c_i) = \pi_i c_i + \sum_{\tau=1}^{T} \max_{r \geq 0} \sum_{j=1}^{n} \lambda P_j(r) \left(r_j - \sum_{k=1}^{m} a_{kj} \pi_k \right)$$
$$= \pi_i c_i + d \max_{r \geq 0} \sum_{j=1}^{n} P_j(r) \left(r_j - \sum_{k=1}^{m} a_{kj} \pi_k \right).$$

It follows that

$$v_{1,i}^*(c_i) + \sum_{k \neq i} c_k \pi_k \leq v_{1,i}(c_i) + \sum_{k \neq i} c_k \pi_k$$
$$= \sum_{k=1}^{m} \pi_k c_k + d \max_{r \geq 0} \sum_{j=1}^{n} P_j(r) \left(r_j - \sum_{k=1}^{m} a_{kj} \pi_k \right)$$
$$= z_{NLP}.$$

This completes the proof. ∎

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 that 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 (2009). 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

$$(\textbf{DP} - \textbf{CHOICE})$$

$$u_t(x) = \max_{S \subseteq N(x)} \left\{ \sum_{j \in S} \lambda P_j(f(S))(f_j + u_{t+1}(x - A^j)) + (\lambda P_0(f(S)) + 1 - \lambda)u_{t+1}(x) \right\}$$

$$= \max_{S \subseteq N(x)} \left\{ \sum_{j \in S} \lambda P_j(f(S))[f_j - (u_{t+1}(x) - u_{t+1}(x - A^j))] \right\} + u_{t+1}(x), \qquad \forall t, x.$$

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. Here N is the set of all products with fares denoted by the vector f. 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 problems due to the curse of dimensionality. Approximate solution approaches are proposed in the literature based on a deterministic 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

$$(\textbf{CDLP}) \qquad z_{CDLP} = \max_{h} \sum_{S \subseteq N} \lambda R(S) h(S)$$

$$\sum_{S \subseteq N} \lambda Q(S) h(S) \leq c \qquad (7)$$

$$\sum_{S \subseteq N} h(S) = T \qquad (8)$$

$$h(S) \geq 0, \quad \forall S \subseteq N.$$

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 (7) 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 twofold. First, we would like to study the computational performance of the decomposition approach to dynamic pricing in the network setting. We also report performance of the decomposition bounds. Second, perhaps more importantly, we would like to compare the performance of dynamic pricing policies to other alternative control strategies.

6.1. 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

$$v_t(x) \approx \sum_{i=1}^{m} v_{t,i}(x_i). \qquad (9)$$

By using (9), 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

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

- **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).

- **NLP5**: This policy implements the optimal prices from (**NLP**), but resolves 5 times with equally spaced resolving intervals.

- **CHOICE**: This policy implements the dynamic programming decomposition introduced in Liu and van Ryzin (2008). It is a dynamic availability control policy.

We have also tried bid-price control policies, where the dual values from CDLP are used as bid-prices. Our numerical results indicate that the policies generate revenues very close to these of static pricing. We therefore choose not to report the results.

We use simulation to evaluate the performance of the different approaches. For each set of instances, we randomly generated 5,000 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.

6.2. Computational Time and Bound Performance

In this section, we report the computational time on randomly generated hub-and-spoke network 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 1 shows a hub-and-spoke network with one hub and 4 non-hub locations. There is 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
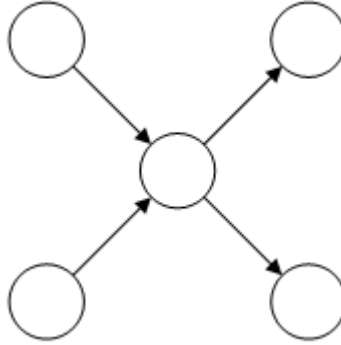
Figure 1: Hub-and-spoke network with 4 locations.

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.

The number of periods is in the set $\{200, 400, 800\}$. The capacity is varied proportionally to the number of time periods. The largest problem instance we consider has 16 non-hub locations and 80 products. 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 a uniform $[0, 20]$ distribution and μ_l is generated from a 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 the X_k's are independent uniform $[0, 1]$ random variables. Note that, once generated, γ_l is held constant throughout the booking horizon.

Table 1 reports the CPU seconds for the different problem instances when solving **DCOMP**. For the largest problem instance, the solution time is about 702 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 2 reports the bounds from **NLP** and **DCOMP**. Table 3 reports the ratio between the **NLP** bound and the decomposition bound. 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 ranges between 1-5%. For large problem instances, the relative difference between the two bounds is relatively small. This is not surprising since the **NLP** bound is asymptotically tight; see Gallego and van Ryzin (1997). Nevertheless, as shown in our simulation results later, the performance lift from **DCOMP** can be quite significant, even when the bounds are close.

6.3. Policy Performance

6.3.1. Problem Instances

We conduct numerical experiments using two sets of randomly generated hub-and-spoke instances. The network structure is similar to the one presented in Figure 1. In the first set of examples, which we call **HS1**, there are 4 non-hub locations, and the total number of periods $T = 500$. There are 4 scheduled flights, each with a capacity of 30, two of which are to the hub, and the other two are from the hub, as shown in Figure 1. In total, there are 4 local itineraries and 4 through itineraries. 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 local itinerary are generated from uniform $[10, 1000]$ and uniform $[10, 100]$ distributions, respectively. 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 values of μ_l and μ_{l0} are generated from uniform $[0, 100]$ distributions. 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 the 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 cases 1 through 10.

Another set of hub-and-spoke example, which we call **HS2**, has 8 non-hub locations. The network topology is essentially the same as **HS1**. The problem data are generated in a similar fashion, except that only one product is offered for each itinerary. There are 8 scheduled flights, each with a capacity of 30, four of which are to the hub, and the other four are from the hub. The number of periods is 1000, and the total number of products is

24. The u_{lj} values for local itineraries 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 values of μ_l and μ_{l0} are generated from uniform $[0, 100]$ and uniform $[0, 20]$ distributions, respectively. All other parameters are generated in the same way as for **HS1**. Ten different problem instances, labeled cases 1 through 10 are generated.

6.3.2. Results for HS1

Table 4 reports the simulated average revenues for the policies we consider. We also report the bounds from dynamic programming decomposition and the deterministic approximation. 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 1-4%. It should be pointed out that the decomposition bound is tighter than the bound from **NLP**, confirming our analytical results in Proposition 4. We also compare the performance of **DCOMP** to **STATIC**, **NLP5**, and **CHOICE**. Observe that **CHOICE** is not performing as well as **STATIC** in almost all problem instances. This is quite surprising, given that **CHOICE** is a dynamic capacity-dependent policy, 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, **STATIC** does not perform as well as the dynamic pricing strategy **DCOMP**. Indeed, **DCOMP** shows a consistent 3-6% revenue improvement across the board. In most RM settings, this improvement is quite significant. A comparison between **NLP5** and **DCOMP** shows that the former performs worse, even though it is re-optimized a few times throughout the booking horizon. This shows that the dynamic pricing strategy should be considered when possible in practice. When the dynamic pricing strategy is not feasible, **STATIC** provides a very strong heuristic, considering its strong performance and its static nature.

6.3.3. Results for HS2

Table 5 reports the results for **HS2**. The sub-optimality gap of **DCOMP** is slightly larger at 3-5%. The policy **STATIC** performs better than **CHOICE** in the majority of problem instances, confirming the robustness of the policy observed in **HS1**. The dynamic pricing policy **DCOMP** shows significant revenue improvement, up to 6% against the three alter-

native policies. The difference between **DCOMP** and **NLP5** is smaller, but can still be considered as practically significant. Overall, the observations are in line with those for **HS1**.

It is also of considerable interest to look at the price path of different products. Figure 2 shows the price path of four different products over 100 periods for one problem instance in **HS2** for **DCOMP**. To obtain a cleaner picture, we did not show the price path for all products over the whole booking horizon (1000 time periods), but the pattern looks very similar. We note that pricing decisions need to balance two aspects of the problem. First, as time goes by, there are less opportunities to sell, and therefore prices may go down. On the other hand, prices may go up since capacities may be consumed over time and become scarce as sales take place. Indeed, Figure 2 shows that prices may go up or down over time, presumably reflecting these two aspects. It is also interesting to note that price changes in the figure tend to be slow. This should be contrasted with choice-based availability control, where each product is either available at a fixed price or is not available (which is conceptually equivalent to charging an extremely high price). We believe the relative strength of **DCOMP** over **CHOICE** comes from this finer control of pricing.

7. Summary and Future Directions

This paper studies the value of dynamic pricing by comparing it with several other reasonable RM approaches, including static pricing and choice-based availability control. Our results show that dynamic pricing can lead to a significant across the board revenue lift, in the order of 1-6% in our numerical study. 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. We also show that dynamic programming decomposition leads to an upper bound on revenue, which is provably tighter than the bound from a deterministic approximation.

Our research suffers from the following limitation. First of all, the static prices considered in this research were generated from a deterministic approximation, which ignores demand uncertainty. Because of this, the gap reported in this paper between dynamic and static pricing may be an overestimate of the true gap between the two. In the same vein, the fixed prices fed to the choice-based availability control were sub-optimal. Future research will benefit from more realistic modeling of static pricing.
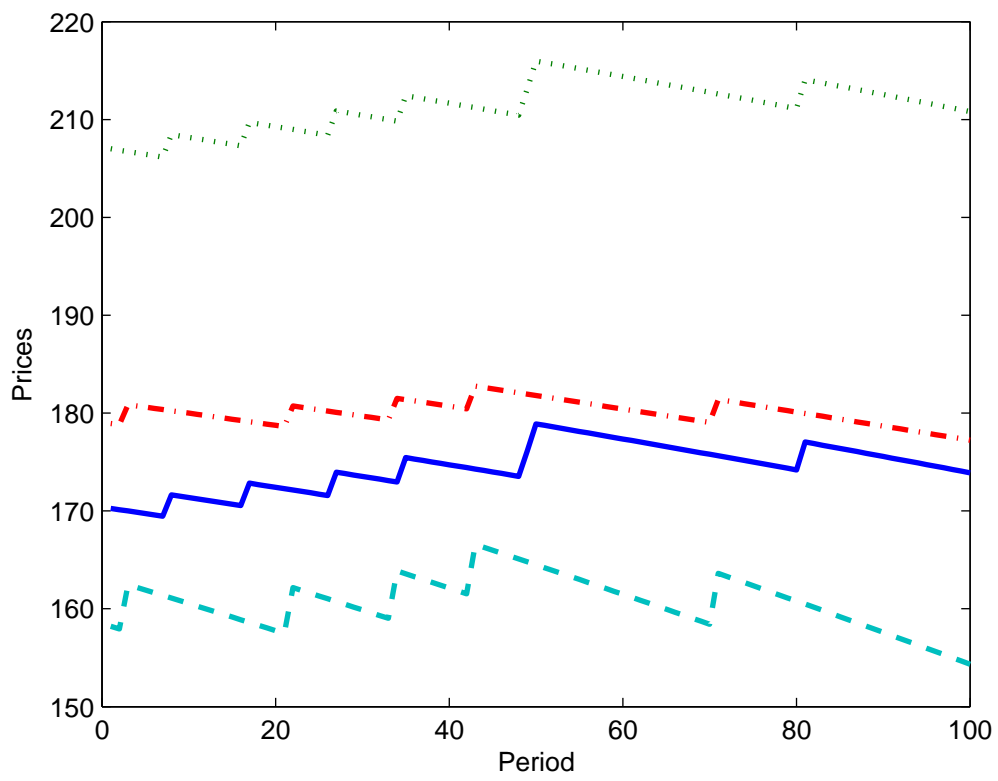
Figure 2: Price path for four different products.

Acknowledgment

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.

Aydin, G., J.K. Ryan. 2002. Product line selection and pricing under the multinomial logit choice model. Working Paper, Purdue University.

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.

Erdelyi, A., H. Topaloglu. 2010. Using decomposition methods to solve pricing problems in network revenue management. Forthcoming, *Journal of Revenue and Pricing Management*.

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.

Karaesman, I., G. J. van Ryzin. 2004. Coordinating overbooking and capacity control decisions on a network. Working paper, Columbia University, New York.

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

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

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

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

non-hub locations, # resources, # products

τ	2,2,3 capacity per leg	NLP seconds	DCOMP seconds	4,4,8 capacity per leg	NLP seconds	DCOMP seconds	8,8,24 capacity per leg	NLP seconds	DCOMP seconds	16,16,80 capacity per leg	NLP seconds	DCOMP seconds
200	20	0.17	2.13	10	0.30	4.84	5	0.72	13.66	2	3.44	35.33
400	40	0.11	8.41	20	0.53	19.30	10	1.59	54.05	5	2.95	175.61
800	80	0.13	33.72	40	0.41	77.34	20	2.69	215.73	10	9.08	702.14

Table 1: CPU seconds for **DCOMP** in hub-and-spoke test cases.

non-hub locations, # resources, # products

τ	2,2,3 capacity per leg	NLP bound	DCOMP bound	4,4,8 capacity per leg	NLP bound	DCOMP bound	8,8,24 capacity per leg	NLP bound	DCOMP bound	16,16,80 capacity per leg	NLP bound	DCOMP bound
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 2: Bounds from **NLP** and **DCOMP** in hub-and-spoke test cases.

non-hub locations, # resources, # products

τ	2,2,3 capacity per leg	bound ratio	4,4,8 capacity per leg	bound ratio	8,8,24 capacity per leg	bound ratio	16,16,80 capacity per leg	bound ratio
200	20	1.04	10	1.04	5	1.04	2	1.05
400	40	1.02	20	1.02	10	1.03	5	1.02
800	80	1.01	40	1.01	20	1.01	10	1.01

Table 3: The ratio of **NLP** and **DCOMP** bounds in hub-and-spoke test cases.

Table 4 (HS1):

Case number	STATIC REV	NLP5 REV	CHOICE REV	DCOMP REV	DCOMP Revenue Gains			OPT-GAP	DCOMP Bound	NLP Bound	Bound difference
					%STATIC	%NLP5	%CHOICE				
1	14534.25	14962.38	14471.35	15174.13	4.22%	1.40%	4.63%	-2.55%	15570.60	15853.44	1.82%
2	20340.71	21220.54	20126.49	21610.49	5.88%	1.80%	6.87%	-1.94%	22038.00	22281.68	1.11%
3	30838.47	30588.69	30662.42	31996.54	3.62%	4.40%	4.17%	-3.14%	33033.45	33793.69	2.30%
4	18274.24	18893.16	18142.28	19093.89	4.29%	1.05%	4.98%	-2.82%	19647.95	19931.89	1.45%
5	21945.55	22805.55	21713.05	23072.22	4.88%	1.16%	5.89%	-3.05%	23797.40	24145.96	1.46%
6	37359.17	37507.72	37180.55	39656.35	5.79%	5.42%	6.24%	-2.24%	40565.83	40986.16	1.04%
7	17535.10	17546.41	17450.33	18441.34	4.91%	4.85%	5.37%	-2.67%	18946.84	19210.81	1.39%
8	21448.97	21779.94	21313.10	22447.77	4.45%	2.98%	5.05%	-3.06%	23157.51	23578.96	1.82%
9	27015.03	27700.37	26641.76	28523.55	5.29%	2.89%	6.60%	-2.72%	29321.11	29707.53	1.32%
10	30648.19	31781.17	30497.23	32568.43	5.90%	2.42%	6.36%	-1.65%	33115.89	33456.72	1.03%

Table 4: Simulation results for **HS1** (hub-and-spoke network with 4 non-hub locations).

Table 5 (HS2):

Case number	STATIC REV	NLP5 REV	CHOICE REV	DCOMP REV	DCOMP Revenue Gains			OPT-GAP	DCOMP Bound	NLP Bound	Bound difference
					%STATIC	%NLP5	%CHOICE				
1	15108.11	15589.39	15098.79	15735.41	3.99%	0.93%	4.05%	-4.30%	16442.07	16668.13	1.37%
2	14518.55	14862.96	14535.18	15044.22	3.49%	1.20%	3.38%	-4.25%	15712.42	15909.79	1.26%
3	14841.85	15271.92	14818.96	15388.29	3.55%	0.76%	3.70%	-4.10%	16046.92	16274.08	1.42%
4	15654.85	16115.25	15569.98	16208.89	3.42%	0.58%	3.94%	-4.45%	16963.83	17149.68	1.10%
5	15534.62	15877.32	15459.66	16060.99	3.28%	1.14%	3.74%	-3.97%	16724.98	16969.83	1.46%
6	16944.43	17491.37	16919.24	17587.11	3.65%	0.54%	3.80%	-4.56%	18427.34	18651.60	1.22%
7	13758.76	14148.56	13786.75	14290.03	3.72%	0.99%	3.52%	-4.25%	14924.66	15068.08	0.96%
8	16624.78	17227.54	16638.68	17349.69	4.18%	0.70%	4.10%	-4.68%	18201.12	18366.36	0.91%
9	15787.35	16243.12	15430.19	16388.30	3.67%	0.89%	5.85%	-4.40%	17142.61	17366.11	1.30%
10	16889.69	17332.28	16470.92	17470.62	3.33%	0.79%	5.72%	-3.77%	18154.75	18419.15	1.46%

Table 5: Simulation results for **HS2** (hub-and-spoke network with 8 non-hub locations).