Proceedings of the 2009 Winter Simulation Conference
M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin, and R. G. Ingalls, eds.

REVENUE MANAGEMENT: MODELS AND METHODS

Kalyan T. Talluri

ICREA and Universitat Pompeu Fabra
Department of Economics and Business
Jaume I Building, Ramon Trias Fargas, 25-27
08005 Barcelona, SPAIN

Garrett J. van Ryzin

Columbia University
Graduate School of Business
3022 Broadway,
New York, NY 10027, U.S.A.

Itir Z. Karaesmen

University of Maryland
Robert H. Smith School of Business
4357 Van Munching Hall
College Park, MD 20742, U.S.A.

Gustavo J. Vulcano

New York University
Leonard N. Stern School of Business
44 West 4th Street, Suite 8-76
New York, NY 10012, U.S.A.

ABSTRACT

Revenue management is the collection of strategies and tactics firms use to scientifically manage demand for their products and services. The practice has grown from its origins in airlines to its status today as a mainstream business practice in a wide range of industry areas, including hospitality, energy, fashion retail, and manufacturing. This article provides an introduction to this increasingly important subfield of operations research, with an emphasis on use of simulation. Some of the contents are based on excerpts from the book *The Theory and Practice of Revenue Management* (Talluri and van Ryzin 2004a), written by the first two authors of this article.

1 INTRODUCTION

Every seller of a product or service faces a number of fundamental decisions. You want to sell at a time when market conditions are most favorable, but who knows what the future might hold? You want the price to be right—not so high that you put off potential buyers and not so low that you lose out on potential profits. You would like to know how much buyers value your product, but more often than not you must just guess at this number. Businesses face even more complex selling decisions. For example, how can a firm segment buyers by providing different conditions and terms of trade that profitably exploit their different buying behavior or willingness to pay? How can a firm design products to prevent cannibalization across segments and channels? Once it segments customers, what prices should it charge each segment? If the firm sells in different channels, should it use the same price in each channel? How should prices be adjusted over time based on seasonal factors and the observed demand to date for each product? If a product is in short supply, to which segments and channels should it allocate the products? How should a firm manage the pricing and allocation decisions for products that are complements (seats on two connecting airline flights) or substitutes (different car categories for rentals)?

Revenue management (RM) is concerned with the methodology and systems required to make *demand-management* decisions, which can be categorized into (i) Structural decisions: Which selling format to use (such as posted prices, negotiations or auctions); which segmentation or differentiation mechanisms to use (if any); which terms of trade to offer (including volume discounts and cancelation or refund options); how to bundle products; and so on. (ii) Price decisions: How to set posted prices, individual-offer prices, and reserve prices (in auctions); how to price across product categories; how to price over time; how to markdown (discount) over the product lifetime; and so on. (iii) Quantity decisions: Whether to accept or reject an offer to buy; how to allocate output or capacity to different segments, products or channels; when to withhold a product from the market and sale at later points in time; and so on.

978-1-4244-5771-7/09/$26.00 ©2009 IEEE

Which of these decisions is most important in any given business depends on the context. The timescale of the decisions varies as well. Structural decisions about which mechanism to use for selling and how to segment and bundle products are normally strategic decisions taken relatively infrequently. Firms may also have to commit to certain price or quantity decisions, for example, by advertising prices in advance or deploying capacity in advance, which can limit their ability to adjust price or quantities on a tactical level. The ability to adjust quantities may also be a function of the technology of production—the flexibility of the supply process and the costs of reallocating capacity and inventory. For example, the use of capacity controls as a tactic in airlines stems largely from the fact that the different "products" an airline sells (different ticket types sold at different times and under different terms) are all supplied using the same, homogeneous seat capacity. This gives airlines tremendous quantity flexibility, so quantity control is a natural tactic in this industry. Retailers, in contrast, often commit to quantities (initial stocking decisions) but have more flexibility to adjust prices over time.

We qualify RM as being either *quantity-based RM* or *price-based RM* if it uses (inventory- or) capacity-allocation decisions or prices as the primary tactical tool respectively for managing demand. Both the theory and practice of RM differ depending on which control variable is used. The number of topics the field spans is too large to cover adequately in a single article like this. The book, *The Theory and Practice of Revenue Management* (Talluri and van Ryzin 2004a) provides in depth coverage of both quantity and price-based RM as well as supporting topics such as demand modeling, economics, forecasting, and system implementation. Here, we only introduce quantity-based RM and discuss the use of simulation in this area.

2 SINGLE-RESOURCE CAPACITY CONTROL

In this section, we examine some basic results on the problem of quantity-based RM for a single resource; specifically, optimally allocating capacity of a resource to different classes of demand. Two prototypical examples are controlling the sale of different fare classes on a single flight leg of an airline and the sale of hotel rooms for a given date at different rate classes. This is to be contrasted with the multiple-resource—or network—problems of Section 3, in which customers require a bundle of different resources (such as two connecting flights or a sequence of nights at the same hotel).

We assume that the firm sells its capacity in n distinct classes (in the case of airlines, these are called *fare classes*) that require the same resource. Classes are ranked from 1 to n in decreasing order of their revenue values, with $r_1 > r_2 > \cdots > r_n$. In the airline and hotel context, these classes represent different discount levels with differentiated sale conditions and restrictions. The units of capacity are assumed homogeneous, and customers demand a single unit of capacity for the resource. The central problem of this section is how to optimally allocate the capacity of the resource to the various classes. This allocation must be done dynamically as demand materializes and with considerable uncertainty about the quantity or composition of future demand.

2.1 Types of Controls

In the travel industry, reservation systems provide different mechanisms for controlling availability. These mechanisms are usually deeply embedded in the software logic of the reservation system and, as a result, can be quite expensive and difficult to change. Therefore, the control mechanisms chosen for a given implementation are often dictated by the reservation system.

The first type of control are *booking limits* that ration the amount of capacity that can be sold to any particular class at a given point in time. For example, a booking limit of 18 on class 2 indicates that at most 18 units of capacity can be sold to customers in class 2. Beyond this limit, the class would be "closed" to additional class 2 customers. This limit of 18 may be less than the physical capacity, for example, when we protect capacity for future demand from class 1 customers. Booking limits are either *partitioned* or *nested*: A *partitioned booking limit* divides the available capacity into separate blocks (or *buckets*)—one for each class—that can be sold only to the designated class. With a *nested booking limit*, the capacity available to different classes overlaps in a hierarchical manner—with higher-ranked classes having access to all the capacity reserved for lower-ranked classes (and perhaps more). Let the nested booking limit for class j be denoted b_j. Then b_j is the maximum number of units of capacity we are willing to sell to classes j to n. So, naturally, b_1 is equal to the capacity. Effectively, nesting logic simply allows any capacity "left over" after selling to classes of lower ranks to become available for sale to classes of higher rank. Nesting booking limits in this way avoids the problem of capacity being simultaneously unavailable for a higher-ranked class yet available for lower-ranked classes. Most reservations systems that use booking-limit controls quite sensibly use nested rather than partitioned booking limits for this reason.

The second type of control are *protection levels* that specify an amount of capacity to reserve (protect) for a particular class or set of classes. Again, protection levels can be *nested* or *partitioned*. A partitioned protection level is trivially equivalent to a partitioned booking limit. In the nested case, protection levels are again defined for sets of classes—ordered in a hierarchical manner according to class order. The protection level j, denoted y_j, is defined as the amount of capacity to save for classes $j, j-1, \ldots, 1$ combined. The booking limit for class j, b_j is simply the capacity minus the protection level for classes $j-1$ and higher. That is,

$$b_j = C - y_{j-1}, \quad j = 2, \ldots, n,$$

where C is the capacity. For convenience, we define $b_1 = C$ (the highest class has a booking limit equal to the capacity) and $y_n = C$ (all classes combined have a protection level equal to capacity).

The third type of controls are *bid-price controls*. What distinguishes bid-price controls from both booking limits and protection levels is that they are revenue-based rather than class-based controls. Specifically, a bid-price control sets a threshold price (which may depend on variables such as the remaining capacity or time), such that a request is accepted if its revenue exceeds the threshold price and rejected if its revenue is less than the threshold price. Bid-price controls are, in principle, simpler than booking-limit or protection-level controls because they require only storing a single threshold value at any point in time—rather than a set of capacity numbers, one for each class. But to be effective, bid prices must be updated after each sale—and possibly also with time as well—and this typically requires storing a table of bid price values indexed by the current available capacity, current time, or both.

2.2 Static Models

In this section, we examine one of the first models for quantity-based RM, the so-called *static* single-resource model. The static model makes several simplifying assumptions that are worth examining in some detail. The first is that demand for the different classes arrives in nonoverlapping intervals in the order of increasing prices of the classes. This could be justified by the observation that advance-purchase discount demand typically arrives before full-fare coach demand in the airline case. Moreover, the optimal controls that emerge from the model can be applied—at least heuristically—even when demand comes in arbitrary order. As for the strict low-before-high assumption, this represents something of a worst-case scenario; for instance, if high-revenue demand arrives before low-revenue demand, the problem is trivial because we simply accept demand first come, first serve.

The second main assumption is that the demands for different classes are independent random variables. Largely, this assumption is made for analytical convenience because dealing with dependence in the demand structure would require introducing complex state variables on the history of observed demand. A third assumption is that demand for a given class does not depend on the capacity controls; in particular, it does not depend on the availability of other classes. However, in practice, customers in a high revenue class may buy down to a lower class if the latter is available, and customers in a lower class may buy up to a higher class if the lower class is closed.

A fourth assumption in the static model is that it suppresses many details about the demand and control process within each of the periods. However, the form of the optimal control is not sensitive to this assumption. A fifth assumption of the model is that either there are no groups, or if there are group bookings, they can be partially accepted.

Finally, the static models assume risk-neutrality. This is a reasonable assumption in practice, since a firm implementing RM typically makes such decisions for a large number of products sold repeatedly (for example, daily flights or daily hotel room stays). Maximizing the average revenue, therefore, is what matters in the end. While we do not cover this case here, some researchers have recently analyzed the single-resource problem with risk-averse decision makers or using worst-case analysis, for example, Lan et al. (2007).

We start with the simple two-class model in order to build some basic intuition and then examine the more general n-class case.

2.2.1 Littlewood's Two-Class Model

The earliest single-resource model for quantity-based RM is due to Littlewood (1972). The model assumes two product classes, with associated prices $r_1 > r_2$. The available capacity is C. Demand for class j is denoted D_j, and its distribution is denoted by $F_j(\cdot)$. Demand for class 2 arrives first. The problem is to decide how much class 2 demand to accept before seeing the realization of class 1 demand.

The optimal decision for this two-class problem can be derived informally using a simple marginal analysis: Suppose that we have x units of capacity remaining and we receive a request from class 2. If we accept the request, we collect revenues of r_2. If we do not accept it, we will sell unit x (the marginal unit) at r_1 if and only if demand for class 1 is x or higher. That is, if and only if $D_1 \geq x$. Thus, the expected gain from reserving the x^{th} unit for class 1 (the *expected marginal value*) is $r_1 P(D_1 \geq x)$. Therefore, it makes sense to accept a class 2 request as long as its price exceeds this marginal value, or equivalently, if and only if

$$r_2 \geq r_1 P(D_1 \geq x). \tag{1}$$

Note the right-hand side of (1) is decreasing in x. Therefore, there will be an optimal protection level, denoted y_1^*, such that we accept class 2 if the remaining capacity exceeds y_1^* and reject it if the remaining capacity is y_1^* or less. Formally, y_1^* satisfies

$$r_2 < r_1 P(D_1 \geq y_1^*) \quad \text{and} \quad r_2 \geq r_1 P(D_1 \geq y_1^* + 1).$$

If a continuous distribution $F_1(x)$ is used to model demand (as is often the case), then the optimal protection level y_1^* is given by the simpler expressions

$$r_2 = r_1 P(D_1 > y_1^*), \quad \text{equivalently,} \quad y_1^* = F_1^{-1}(1 - \frac{r_2}{r_1}), \tag{2}$$

which is known as *Littlewood's rule*. Setting a protection level of y_1^* for class 1 according to Littlewood's rule is an optimal policy. Equivalently, setting a booking limit of $b_2^* = C - y_1^*$ on class 2 demand is optimal. Alternatively, we can use a bid-price control with the bid price set at $\pi(x) = r_1 P(D_1 > x)$.

2.2.2 n-Class Models

We next consider the general case of $n > 2$ classes. Again, we assume that demand for the n classes arrives in n stages, one for each class, with classes arriving in increasing order of their revenue values. Hence, class n (the lowest price) demand arrives in the first stage (stage n), and the highest price class (class 1) arrives in the last stage (stage 1). Since there is a one-to-one correspondence between stages and classes, we index both by j. Demand and capacity are most often assumed to be discrete, but occasionally we model them as continuous variables when it helps simplify the analysis and optimality conditions.

Dynamic Programming Formulation: This problem can be formulated as a dynamic program in the stages (equivalently, classes), with the remaining capacity x being the state variable. At the start of each stage j, the demand $D_j, D_{j-1}, \ldots, D_1$ has not been realized. Within stage j, the model assumes that the following sequence of events occurs:

1. The realization of the demand D_j occurs, and we observe its value.
2. We decide on a quantity u of this demand to accept. The amount accepted must be less than the capacity remaining, so $u \leq x$. The optimal control u^* is therefore a function of the stage j, the capacity x, and the demand D_j, $u^* = u^*(j, x, D_j)$, though we often suppress this explicit dependence on j, x and D_j in what follows.
3. The revenue $r_j u$ is collected, and we proceed to the start of stage $j-1$ with a remaining capacity of $x - u$.

This sequence of events is assumed for analytical convenience; we derive the optimal control u^* "as if" the decision on the amount to accept is made *after* knowing the value of demand D_j. In reality, of course, demand arrives sequentially over time, and the control decision has to be made *before* observing all the demand D_j. However, it turns out that optimal decisions do not use the prior knowledge of D_j as we show below. Hence, the assumption that D_j is known is not restrictive.

Let $V_j(x)$ denote the value function at the start of stage j. Once the value D_j is observed, the value of u is chosen to maximize the current stage j revenue plus the revenue to go, or

$$r_j u + V_{j-1}(x - u),$$

subject to the constraint $0 \leq u \leq \min\{D_j, x\}$. The value function entering stage j, $V_j(x)$, is then the expected value of this optimization with respect to the demand D_j. Hence, the Bellman equation is

$$V_j(x) = E\left[\max_{0 \leq u \leq \min\{D_j, x\}} \{r_j u + V_{j-1}(x - u)\}\right], \tag{3}$$

with boundary conditions

$$V_0(x) = 0, \quad x = 0,1,\ldots,C.$$

Note that the Bellman equation is of the form $E[\max\{\cdot\}]$ and not $\max E[\cdot]$ as in many standard texts. But essentially, the $\max E[\cdot]$ form can be recovered by considering the demand D_j to be a state variable along with x. While the two forms can be shown to be equivalent, the $E[\max\{\cdot\}]$ is simpler to work with in many RM problems. In our case, this leads to the modeling assumption that we optimize "as if" we observed D_j. The values u^* that maximize the right-hand side of (3) for each j and x form an optimal control policy for this model.

Optimal Policy with Discrete Demand and Capacity: We first consider the case where demand and capacity are discrete. To analyze the form of the optimal control in this case, define

$$\Delta V_j(x) \equiv V_j(x) - V_j(x-1).$$

$\Delta V_j(x)$ is the *expected marginal value of capacity* at stage j—the expected incremental value of the x^{th} unit of capacity. A key result concerns how these marginal values change with capacity x and the stage j:

Proposition 1 *The marginal values $\Delta V_j(x)$ of the value function $V_j(x)$ defined by (3) satisfy $\forall x, j$: (i) $\Delta V_j(x+1) \leq \Delta V_j(x)$, and (ii) $\Delta V_{j+1}(x) \geq \Delta V_j(x)$.*

That is, at a given stage j the marginal value is decreasing in the remaining capacity, and at a given capacity level x the marginal value increases in the number of stages remaining. These two properties are intuitive and greatly simplify the control. The resulting optimal control can be expressed in terms of optimal protection levels y_j^* for $j, j-1,\ldots,1$ (class j and higher in the revenue order) by

$$y_j^* \equiv \max\{x : r_{j+1} < \Delta V_j(x)\}, \quad j = 1,\ldots,n-1. \tag{4}$$

(Recall the optimal protection level $y_n^* \equiv C$ by convention.) The optimal control at stage $j+1$ is then $u^*(j+1,x,D_{j+1}) = \min\{(x-y_j^*)^+, D_{j+1}\}$, where the notation $z^+ = \max\{0,x\}$ denotes the positive part of z. The quantity $(x-y_j^*)^+$ is the remaining capacity in excess of the protection level, which is the maximum capacity we are willing to sell to class $j+1$.

In practice, we can simply post the protection level y_j^* in a reservation system and accept requests first come, first serve until the capacity threshold y_j^* is reached or the stage ends, whichever comes first. Thus, the optimal protection-level control at stage $j+1$ requires no information about the demand D_{j+1}, yet it produces the same optimal decision "as if" we knew D_{j+1} exactly at the start of stage $j+1$. The reason for this is that knowledge of D_{j+1} does not affect the future value of capacity, $V_j(x)$. Deciding to accept or reject each request simply involves comparing current revenues to the marginal value of capacity, and this comparison does not depend on how many stage-$(j+1)$ requests there are in total.

Proposition 1(ii) implies the nested protection structure

$$y_1^* \leq y_2^* \leq \cdots \leq y_n^*.$$

One can also use booking limits in place of protection levels to achieve the same control. Optimal nested booking limits are defined by

$$b_j^* \equiv C - y_{j-1}^*, \quad j = 2,\ldots,n, \tag{5}$$

with $b_1^* \equiv C$. The optimal control in stage $j+1$ is then to accept

$$u^*(j+1,x,D_{j+1}) = \min\{(b_{j+1} - (C-x))^+, D_{j+1}\}.$$

Note that $C-x$ is the total capacity sold prior to stage $j+1$ and b_{j+1} is the booking limit for class $j+1$, so $(b_{j+1}-(C-x))^+$ is the remaining capacity available for class $j+1$.

Finally, the optimal control can also be implemented through a table of bid prices. Indeed, if we define the stage $j+1$ bid price by

$$\pi_{j+1}(x) \equiv \Delta V_j(x), \tag{6}$$

then the optimal control is

$$u^*(j+1,x,D_{j+1}) = \begin{cases} 0 & \text{if } \ r_{j+1} < \pi_{j+1}(x) \\ \max\{z : r_{j+1} \geq \pi_{j+1}(x-z)\} & \text{otherwise.} \end{cases}$$

In words, we accept the z^{th} request in stage $j+1$ if the price r_{j+1} exceeds the bid price value $\pi_{j+1}(x-z)$ of the z^{th} unit of capacity that is allocated. In practice, we can store a table of bid prices and process requests by sequentially comparing the price of each product to the table values corresponding to the remaining capacity. We summarize these results in the following theorem.

Theorem 1 *For the static model defined by (3), the optimal control can be achieved using either (i) nested protection levels defined by (4), (ii) nested booking limits defined by (5), or (iii) bid-price tables defined by (6).*

Optimality Conditions for Continuous Demand: When the capacity is continuous and demand at each stage has a continuous distribution, the dynamic program is still given by (3); however D_j, x, and u are now continuous quantities. The analysis of the dynamic program is slightly more complex than it is in the discrete-demand case, but many of the details are quite similar. Hence, we only briefly describe the key differences.

The main change is that the marginal value $\Delta V_j(x)$ is now replaced by the derivative of $V_j(x)$ with respect to x, $\frac{\partial}{\partial x}V_j(x)$. This derivative is still interpreted as the marginal expected value of capacity. The marginal value $\frac{\partial}{\partial x}V_j(x)$ is shown to be decreasing in x (equivalently, $V_j(x)$ is concave in x). Therefore, the optimal control in stage $j+1$ is to keep increasing u (keep accepting demand) as long as

$$r_{j+1} \geq \frac{\partial}{\partial x}V_j(x-u)$$

and to stop accepting once this condition is violated or the demand D_{j+1} is exhausted, whichever comes first. Again, this decision rule can be implemented with optimal protection levels, defined by

$$y_j^* \equiv \max\left\{x : r_{j+1} < \frac{\partial}{\partial x}V_j(x)\right\}, \quad j = 1,\ldots,n-1.$$

One of the virtues of the continuous model is that it leads to simplified expressions for the optimal vector of protection levels $\mathbf{y}^* = (y_1^*,\ldots,y_n^*)$. We state the basic result here; see Brumelle and McGill (1993) for a proof.

First, for an arbitrary vector of protection levels $\mathbf{y}$ and vector of demands $\mathbf{D} = (D_1,\ldots,D_n)$, define the following $n-1$ *fill events*

$$B_j(\mathbf{y},\mathbf{D}) \quad \equiv \quad \{D_1 > y_1, D_1+D_2 > y_2, \ldots, D_1+\cdots+D_j > y_j\}, \quad j = 1,\ldots,n-1.$$

$B_j(\mathbf{y},\mathbf{D})$ is the event that demand to come in stages $1,2,\ldots,j$ exceeds the corresponding protection levels. A necessary and sufficient condition for $\mathbf{y}^*$ to be an optimal vector of protection levels is that it satisfy the $n-1$ equations

$$P(B_j(\mathbf{y}^*,\mathbf{D})) = \frac{r_{j+1}}{r_1}, \quad j = 1,2,\ldots,n-1. \tag{7}$$

That is, the j^{th} fill event should occur with probability equal to the ratio of class $(j+1)$ revenue to class 1 revenue. As it should, this reduces to Littlewood's rule (2) in the $n=2$ case, since $P(B_1(\mathbf{y}^*,\mathbf{D})) = P(D_1 > y_1^*) = r_2/r_1$. Note that

$$B_j(\mathbf{y},\mathbf{D}) = B_{j-1}(\mathbf{y},\mathbf{D}) \cap \{D_1+\cdots+D_j > y_j\},$$

so the event $B_j(\mathbf{y},\mathbf{D})$ can occur only if $B_{j-1}(\mathbf{y},\mathbf{D})$ occurs. Also, if $y_j = y_{j-1}$ then $B_j(\mathbf{y},\mathbf{D}) = B_{j-1}(\mathbf{y},\mathbf{D})$. Thus, if $r_j < r_{j-1}$, we must have $y_j^* > y_{j-1}^*$ to satisfy (7). Thus, the optimal protection levels are strictly increasing in j if the revenues are strictly decreasing in j.

Computing Optimal Protection Levels: One approach in computing the optimal protection levels in the n-class static model is based on using dynamic programming recursion (3) directly and requires that demand and capacity are discrete - or in the continuous case that these quantities can be suitably discretized. A second approach is based on using (7) together with

Monte Carlo integration. This is most suitable for the case of continuous demand and capacity, though the discrete case can be computed (at least heuristically) with this method as well. The idea is to simulate K demand vectors $\mathbf{d}^k = (d_1^k, ..., d_n^k)$, $k = 1, .., K$, from the forecast distributions for the n classes. We then sort through these values to find thresholds $\mathbf{y}$ that approximately satisfy (7). Note that

$$P(B_j(\mathbf{y}, \mathbf{D})) = P(\sum_{i=1}^{j} D_i > y_j | B_{j-1}(\mathbf{y}, \mathbf{D})) P(B_{j-1}(\mathbf{y}, \mathbf{D})).$$

Thus, (7) implies that $\mathbf{y}^*$ must satisfy

$$P(\sum_{i=1}^{j} D_i > y_j^* | B_{j-1}(\mathbf{y}^*, \mathbf{D})) = \frac{1}{P(B_{j-1}(\mathbf{y}^*, \mathbf{D}))} \frac{r_{j+1}}{r_1} = \frac{r_{j+1}}{r_j}$$

for $j = 1, ..., n-1$. The following algorithm, suggested by Robinson (1995), computes the optimal $\mathbf{y}^*$ approximately using the empirical conditional probabilities estimated from the sample of simulated demand data:

STEP 0: Generate and store K random demand vectors $\mathbf{d}^k = (d_1^k, ..., d_n^k)$. For $k = 1, .., K$ and $j = 1, ..., n-1$, compute the partial sums $S_j^k = \sum_{i=1}^{j} d_i^k$ and form the vector $\mathbf{S}^k = (S_1^k, ..., S_{n-1}^k)$. Initialize a list $\Upsilon = \{1, ..., K\}$ and counter $j = 1$.

STEP 1: Sort the vectors $\mathbf{S}^k$, $k \in \Upsilon$ by their j^{th} component values, S_j^k. Let $[l]$ denote the l^{th} element of Υ in this sorted list so that

$$S_j^{[1]} \leq S_j^{[2]} \cdots \leq S_j^{[|\Upsilon|]}.$$

STEP 2: Set $l^* = \lfloor \frac{r_{j+1}}{r_j} |\Upsilon| \rfloor$. Set $y_j = \frac{1}{2}(S_j^{[l^*]} + S_j^{[l^*+1]})$.

STEP 3: Set $\Upsilon \leftarrow \{k \in \Upsilon : S_j^k > y_j\}$, and $j \leftarrow j+1$. IF $j = n-1$ STOP. ELSE GOTO STEP 1.

The complexity of this method is $O(nK \log K)$, which makes it relatively efficient even with large samples.

Computing Protection Levels with No Demand Information: So far, the models and methods we introduced assume the probability distribution of demand in each class is known. However, in many applications, the revenue manager does not know the demand distribution, either because data is not available, or forecasting is challenging because data is censored (i.e., represents only sales as opposed to true demand). In those cases, optimal protection levels can be computed adaptively, without recourse to the complex cycles of forecasting and optimization. The method of van Ryzin and McGill (2000) updates booking policy parameters from one flight to the next, keeping track of the occurrence of fill events on previous flights. The method relies on the optimality conditions (7) and works for underlying continuous demand distributions. It is provably convergent to an optimal policy with repeated application. This approach can also be used in simulation-based optimization when demand distributions are known because the main idea is to compute stochastic-gradients of the objective function. Kunnumkal and Topaloglu (2008) present a similar stochastic approximation method provably convergent for the case of discrete demand distributions. More recently, Huh and Rusmevichientong (2007) proposed another adaptive method that uses results from online convex optimization theory and works directly with the expected revenue function (3), as opposed to fill events.

2.3 Dynamic Models

Dynamic models relax the assumption that the demand for classes arrives in a strict low-to-high revenue order. Instead, they allow for an arbitrary order of arrival, with the possibility of interspersed arrivals of several classes. While at first this seems like a strict generalization of the static case, the dynamic models require the assumption of Markovian (such as Poisson) arrivals to make them tractable. This puts restrictions on modeling different levels of variability in demand. Indeed, this limitation on the distribution of demand is the main drawback of dynamic models in practice. In addition, dynamic models require an estimate of the pattern of arrivals over time (called the *booking curve*), which may be difficult to calibrate in certain applications. Thus, the choice of dynamic versus static models essentially comes down to a choice of which set of approximations is more acceptable and what data is available in any given application.

Retaining the other assumptions of the static model, one can develop a dynamic programming formulation and determine the *time-based* booking limits, protection levels, or bid prices to optimally control the capacity of the resource.

3 NETWORK PROBLEMS

We next examine the problem of capacity control on a network of resources; for example, managing the capacities of a set of flights in a hub-and-spoke airline network with connecting and local traffic. The dependence among the resources in such cases is created by customer demand; customers may require several resources simultaneously (e.g. two connecting flights) to satisfy their needs. Thus, limiting availability of one resource may cause a loss of demand for complementary resources. This in turn creates dependencies among the resources that necessitates making control decisions at the network level. In the airline industry, network RM is also called "the passenger mix problem" or "O&D (origin-destination) control".

Simulation studies of airline hub-and-spoke networks have shown that there can be significant revenue benefits from using network methods over single-resource methods; see Williamson (1992) and Belobaba (2001). In terms of industrial practice, the potential improvements have been sufficient to justify significant investments in network RM systems within the airline industry, hotel industry, and elsewhere. However, network RM poses significant implementation and methodological challenges. On the implementation side, network RM vastly increases the complexity and volume of data that one must collect, store and manage. On the forecasting side, it requires a massive increase in the scale of the forecasting system, which now must produce forecasts for each individual itinerary and price-class combination - which we will call a *product* - at each point in the booking process. Optimization is more complex as well. In the case of a single-resource problem there are many exact optimization methods as we discussed in Section 2, but in the network case exact optimization is, for all practical purposes, impossible. Therefore optimization methods necessarily require approximations of various types. Achieving a good balance between the quality of the approximation and the efficiency of the resulting algorithms becomes the primary challenge.

3.1 Types of Controls

As with single-resource problems, in network problems there are a variety of ways one can control the availability of capacity. We next look at the major categories of network controls. Most are network versions of the controls used for single-resource problems. But others, virtual nesting in particular, are somewhat unique to the network setting.

3.1.1 Virtual nesting controls

Nested booking limits, of the type we saw in Section 2 for the single-resource case, are difficult to translate directly into a network setting. However, the ability of nested controls to dynamically share the capacity of a resource - and thereby recover the pooling economies lost in partitioned controls - is an attractive feature. Thus, it is desirable to have a control that combines these features.

Virtual nesting control –a hybrid of network and single-resource controls– provides one solution. This control scheme was developed by American Airlines beginning in 1983 as a strategy for incorporating some degree of network control within the single-leg nested allocation structure of American's (then leg-based) reservation systems; see Smith, Leimkuhler and Darrow (1992).

Virtual nesting uses single-resource nested booking controls at each resource in the network. However, the classes used in these nested allocations are not the fare classes themselves. Rather, they are based on a set of *virtual classes*. Products are assigned to a virtual class through a process known as *indexing*. This indexing could be updated over time as network demand patterns change, though typically indexing is not a "real time" process. *Nested booking limits* (or protection levels) for each resource are then computed using these virtual classes.

Virtual nesting has proven to be quite effective and popular in practice, especially in the airline industry. It preserves the booking-class control logic of most airline computer reservation systems (CRS) yet incorporates network displacement cost information. It therefore provides a nice compromise between leg-level and full network O&D control.

3.1.2 Bid-price controls

While nested allocations are difficult to extend directly to networks, network bid-price controls are a simple extension of their single-resource versions described in Section 2. In a network setting, a bid-price control sets a threshold price - or *bid-price* - for each resource in the network. Roughly, this bid-price is an estimate of the marginal cost to the network of consuming the next incremental unit of the resource's capacity. When a request for a product comes in, the revenue of the request is compared to the *sum* of the bid prices of the resources required by the product. If the revenue exceeds the sum of the bid prices, the request is accepted; if not, it is rejected. We refer the reader to Talluri and van Ryzin (2004a) for the origins of bid prices controls and the theoretical properties of bid-price mechanisms.

3.2 A General Network Model

We begin with a basic model of the network allocation problem. The network has m resources which can be used to provide n products. We let $a_{ij} = 1$ if resource i is used by product j and $a_{ij} = 0$ otherwise. Define the *incidence matrix* $A = [a_{ij}]$. Thus, the j-th column of A, denoted A_j, is the incidence vector for product j. We also use the notation $i \in A_j$ to indicate that resource i is used by product j.

The state of the network is described by a vector $\mathbf{x} = (x_1, \ldots, x_m)$ of resource capacities. If product j is sold, the state of the network changes to $\mathbf{x} - A_j$. To simplify our analysis at this stage, we will ignore cancelations and no-shows.

Time is discrete, there are T periods and the index t represents the current time (here, time indices run forwards). Within each time Period t, we assume that at most one request for a product can arrive; that is, the discretization of time is sufficiently fine so that the probability of more than one request is negligible. This assumption can be generalized in many of the results below, but is the simplest case to present. It is analogous to the network version of the dynamic single-resource model.

To make the notation more compact, demand in Period t is modeled as the realization of a *single* random vector $\mathbf{R(t)} = (R_1(t), \ldots, R_n(t))$. If $R_j(t) = r_j > 0$, this indicates a request for product j occurred and that its associated revenue is $R_j(t)$; if $R_j(t) = 0$, this indicates no request for product j occurred. A realization $\mathbf{R(t)} = 0$ (all components equal to zero) indicates that no request from *any* product occurred at time t. For example, if we have $n = 3$ products, then a value $\mathbf{R(t)} = (0,0,0)$ indicates no requests arrived, a value $\mathbf{R(t)} = (120,0,0)$ indicates a request for product 1 with revenue of \$120. Note by our assumption that at most one arrival occurs in each time period, at most one component of $\mathbf{R(t)}$ can be positive (as indicated in the example above). More formally, let $E_n = \{e_0, e_1, \ldots, e_n\}$, where e_j is the j-th unit n-vector and e_0 is the zero n-vector, and define the set $\mathscr{S} = \{R : R = \alpha e, e \in E_n, \alpha \geq 0\}$. Then, $\mathbf{R(t)} \in \mathscr{S}$. The revenue $R_j(t)$ associated with product j may be random as well. The sequence $\{\mathbf{R(t)}; t \geq 1\}$ is assumed to be independent with known joint distributions in each Period t. When revenues associated with product j are fixed, we will also denote these by r_j. We use the notation $\mathbf{R(t)}^\top$ for the transpose of the vector $\mathbf{R(t)}$.

Given the current time, t, the current remaining capacity $\mathbf{x}$ and the current request $\mathbf{R(t)}$, we are faced with a decision: Do we or do we not accept the current request?

Let an n-vector $\mathbf{u(t)}$ denote this decision, where $u_j(t) = 1$ if we accept a request for product j in Period t, and $u_j(t) = 0$ otherwise. In general, the decision to accept, $u_j(t)$, is a function of the remaining capacity vector $\mathbf{x}$ and the revenue r_j of product j, i.e. $u_j(t) = u_j(t, \mathbf{x}, r_j)$, and hence $\mathbf{u(t)} = u(t, \mathbf{x}, \mathbf{r})$. Since we can accept at most one request in any period and resources cannot be oversold, if the current seat inventory is x, then $\mathbf{u(t)}$ is restricted to the set $\mathscr{U}(\mathbf{x}) = \{u \in E_n : Au \leq \mathbf{x}\}$.

3.3 The Structure of the Optimal Controls

In order to formulate a dynamic program to determine optimal decisions $u^*(t, \mathbf{x}, \mathbf{r})$, let $V_t(\mathbf{x})$ denote the maximum expected revenue-to-go given remaining capacity $\mathbf{x}$ in Period t. Then $V_t(\mathbf{x})$ must satisfy the Bellman equation

$$V_t(\mathbf{x}) = E\left[\max_{u \in \mathscr{U}(\mathbf{x})} \left\{\mathbf{R(t)}^\top u(t, \mathbf{x}, \mathbf{R(t)}) + V_{t+1}(\mathbf{x} - Au)\right\}\right], \tag{8}$$

with the boundary condition $V_{T+1}(\mathbf{x}) = 0, \forall \mathbf{x}$.

Therefore, a control u^* is optimal if and only if it satisfies:

$$u_j^*(t, \mathbf{x}, r_j) = \begin{cases} 1 & r_j \geq V_{t+1}(\mathbf{x}) - V_{t+1}(\mathbf{x} - A_j),\ A_j \leq \mathbf{x} \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

The control (9) says that an optimal policy for accepting requests is of the form: accept revenue r_j for product j if and only if we have sufficient remaining capacity and

$$r_j \geq V_{t+1}(\mathbf{x}) - V_{t+1}(\mathbf{x} - A_j),$$

where $R_j(t) = r_j$ is the revenue value of the request for product j. This reflects the rather intuitive notion that we accept a revenue of r_j for product j only when it exceeds the *opportunity cost* of the reduction in resource capacities required to satisfy the request.

One can show that bid prices are not able to achieve the optimal control (9) in all cases due to the non-additive nature of the value function; see Talluri and van Ryzin (1998). At the same time, one can show that bid-price controls have good asymptotic properties and are in fact asymptotically optimal as the number of seats sold and the demand are increased (proportionately), as shown in Talluri and van Ryzin (1998). However, the real test of network control methods and models in practice must be determined through careful simulation testing.

3.4 Approximations Based on Network Models

The formulation (8) cannot be solved exactly for most networks of realistic size. Instead, one must rely on approximations of various types. Most approximation methods proposed to date follow one of two basic (not necessarily mutually exclusive) strategies: The first, which we look at in this subsection, is to use a simplified network model. For example, posing the problem as a static math program. The second strategy, which we look at in Section 3.5, is to decompose the network problem into a collection of single-resource problems. Whichever method is used, it is useful to view all such methods as producing different approximations of the optimal value function. The outputs from these approximations can be used to construct controls of various types - bid-price controls, partitioned or nested allocations, or virtual nesting controls.

Among the most useful information provided by an approximation method are estimates of displacement (opportunity) costs - or bid prices. These are used either directly in bid-price control mechanisms, or indirectly in other mechanisms like virtual nesting. Given an approximation method M that yields an estimate of the value function $V_t^M(\mathbf{x})$, we can approximate the displacement cost of accepting product j by

$$V_t^M(\mathbf{x}) - V_t^M(\mathbf{x} - A_j) \approx \nabla_x^\top V_t^M(\mathbf{x}) A_j,$$

where $\nabla_x V_t^M(\mathbf{x})$ is the gradient of the value function approximation $V_t^M(\mathbf{x})$, assuming the gradient exists. The bid prices are then defined by

$$\pi_i^M(t, \mathbf{x}) = \frac{\partial}{\partial x_i} V_t^M(\mathbf{x})$$

If the gradient does not exist, then $\nabla_x V_t^M(\mathbf{x})$ is typically replaced (at least implicitly) by a subgradient of $V_t^M(\mathbf{x})$. If the approximation is discrete, then first differences are used in place of partial derivatives.

Clearly, one objective for an approximation method is to produce a good estimate of the value function - and more importantly, a good estimate of the displacement costs or bid prices. On the other hand, speed of computation matters as well. The approximation $V_t^M(\mathbf{x})$ may be a static approximation that must be resolved quite frequently in practice to account for changes in remaining capacity $\mathbf{x}$ and remaining time t. A static method that is accurate but very computationally complex will therefore be of little use in practice. Thus, one should always keep these two criteria - accuracy and speed - in mind when judging network approximation methods. We introduce two network approximation methods below.

Deterministic Linear Programming Model: The *deterministic linear programming* (DLP) method uses the approximation

$$V_t^{LP}(\mathbf{x}) = \max_z \mathbf{r}^\top z \quad s.t. \quad Az \leq \mathbf{x}, \quad 0 \leq z \leq E[\mathbf{D}] \tag{10}$$

where recall $\mathbf{D} = (D_1, \ldots, D_n)$ is the vector of demand-to-come (demand over the periods $t, t+1, \ldots, T$) for product j, $j = 1, \ldots, n$, and $\mathbf{r} = (r_1, .., r_n)$ is the vector of revenues associated with the n products. The decision variables $z = (z_1, \ldots, z_n)$ represent partitioned allocation of capacity for each of the n products. The approximation effectively treats demand as if it were deterministic and equal to its mean $E[D]$. Once the DLP is solved, typically its dual solution, not the primal solution, is used: the optimal dual variables, π^{LP}, associated with the constraints $Az \leq \mathbf{x}$ are used as bid prices.

The DLP was among the first models analyzed for network RM; see the references in Talluri and van Ryzin (2004a). The main advantage of the DLP model is that it is computationally very efficient to solve. Due to its simplicity and speed, it is a popular in practice. The weakness of the DLP approximation is that it considers only the mean demand and ignores all other distributional information. Despite this deficiency, simulation studies have shown that with frequent reoptimization, the performance of DLP bid prices can be quite good. In general, the performance of the DLP method (like many network methods) depends heavily on the type of network, the order in which fare products arrive and the frequency of reoptimization.

Randomized Linear Programming Model: *Randomized linear programming* (RLP) approach proposes one way of incorporating stochastic information into the DLP method: replace the expected value $E[\mathbf{D}]$ by the random vector $\mathbf{D}$ itself and the expected value of the resulting optimal solution then forms an approximation to the value function. That is, define

$$H_t(\mathbf{x},\mathbf{D}) = \max_z \ \mathbf{r}^\top z \ \ s.t. \ \ Az \le \mathbf{x}, \quad 0 \le z \le \mathbf{D} \tag{11}$$

The optimal value $H_t(\mathbf{x},\mathbf{D})$ is a random variable. Let $\pi(\mathbf{x},\mathbf{D})$ denote an optimal vector of dual prices for the set of constraints $Az \le \mathbf{x}$, and note that $\pi(\mathbf{x},\mathbf{D})$ is also a random vector.

Next, consider approximating the value function by the expected value of $H_t(\mathbf{x},\mathbf{D})$,

$$V_t^{RLP}(\mathbf{x}) = E[H_t(\mathbf{x},\mathbf{D})]. \tag{12}$$

Note the right hand side corresponds to a "perfect information" approximation, because it reflects a case in which *future* allocations (and revenues) are based on perfect knowledge of the realized demand $\mathbf{D}$. We then use $\nabla_x E[H_t(\mathbf{x},\mathbf{D})]$ as a vector of bid prices.

The RLP approach, proposed by Talluri and van Ryzin (1999), relies on an efficient method to compute $\nabla_x E[H_t(\mathbf{x},\mathbf{D})]$: First, simulate K independent samples of the demand vector, $\mathbf{D}^{(1)},\ldots,\mathbf{D}^{(K)}$, and solve (11) for each sample. Then, estimate the gradient using $\pi^{RLP} = \frac{1}{K}\sum_{i=1}^{K}\pi(\mathbf{x},\mathbf{D}^{(i)})$. That is, simply *average* the dual values from K perfect-information allocation solutions on randomly generated demands. Hence the name *randomized linear programming*.

3.5 Approximations Based on Decomposition

Another strategy for generating network controls is to decompose the problem (approximately) into m single-resource problems, each of which may incorporate some network information, but which are nevertheless independent. Formally, one can think of such a decomposition method as follows: an approximation method M decomposes the network problem into m single-resource models, denoted model $i = 1,\ldots,m$, with value functions $V_t^{M_i}(x_i)$, that depends on the time-to-go t and the remaining capacity x_i of resource i. These may be constructed by incorporating some static, network information into the estimates. Then, the total value function is approximated by

$$V_t^M(\mathbf{x}) = \sum_{i=1}^{m} V_t^{M_i}(x_i).$$

Typically, such approximations are discrete and yield bid prices

$$\pi_i(t,\mathbf{x}) = \Delta V_t^{M_i}(x_i), \qquad i = 1,..,m.$$

where $\Delta V_t^{M_i}(x_i) = V_t^{M_i}(x_i) - V_t^{M_i}(x_i - 1)$ is the usual marginal expected value produced by model i.

Decomposition approximations have several advantages relative to network approximations. First, because they are based on single-resource problems, the displacement costs and bid prices are typically dynamic and can be represented as a table of outputs (in the case of dynamic programming models) or simple formulas (in the case of static approximations). Thus, it is easy to quickly determine the effect of changes in both the remaining time t and remaining capacity $\mathbf{x}$ on the resulting bid prices. This should be contrasted with network models, which must typically be re-solved to determine the effects of such changes. Second, because they are often based on simple, single-resource models, decomposition methods allow for more realistic assumptions, such as discrete demand and capacity, sequential decision making over time and stochastic dynamic demand.

The primary disadvantage of decomposition methods is that in the process of separating the problem by resources, it can be difficult to retain important network effects in the approximations. However, hybrids of the two approaches can be used to try to achieve the benefits of both network and decomposition methods. We now introduce one of the decomposition methods commonly used in practice.

Displacement adjusted virtual nesting (DAVN): While virtual nesting is often viewed as a control strategy - and indeed is used as such in most cases in practice - it can also be viewed as a decomposition approximation to the network value function. Indeed, the marginal values produced by the virtual nesting approximation can be used in a bid-price control scheme which avoids the virtual nesting controls entirely.

DAVN starts with a set of static bid prices - or marginal value estimate - which we denote by $\bar{\pi} = (\bar{\pi}_1, \ldots, \bar{\pi}_m)$. These estimates may be obtained, for example, from one of the network math programming models presented in Section 3.4. Given the bid prices $\bar{\pi}$, one then solves a leg-level problem at each resource i as follows:

First, for all products j that use resource i, a displacement adjusted revenue $\bar{r}_{ij}$ is computed using

$$\bar{r}_{ij} = r_j - \sum_{l \in A_j, l \neq i} \bar{\pi}_l. \tag{13}$$

That is, the revenue of product j on resource i is reduced by the static bid-price values of the other resources used by product j. This adjustment is intended to approximate the net benefit of accepting product j on resource i. Note that the displacement adjusted revenue could be negative. In this case, Produce j is never accepted on resource i, and typically we either eliminates product j from the problem on resource i or (equivalently) set the displacement adjusted revenue value to zero.

The next step is clustering - or indexing. In this step, the displacement adjusted revenue values on each resource are clustered into a specified number $\bar{c}$ of virtual classes - or buckets - denoted $c = 1, .., \bar{c}$. The number of virtual classes, $\bar{c}$, is a design parameter, but is typically on the order of 10. It may also vary across resources. The indexing from product j to Virtual Class c on each resource can be performed using a variety of clustering algorithms. The particular indexing method and clustering criteria are also design decisions and vary from implementation to implementation.

Once the virtual classes are formed, we compute a representative revenue value for each class - usually the demand-weighted average revenue. Then, the distribution of total demand in a virtual class is computed - typically by adding the means and variances of demand-to-come. Next, one solves a multi-class, single-resource problem based on these data. The problem could be solved exactly using the static single-leg model or approximately using heuristics such as the expected marginal seat revenue (EMSR) approach of Belobaba (1989). We call this model $DAVN_i$. This procedure yields a set of booking limits (or protection levels) for the virtual classes at each resource i and a value function estimate $V_t^{DAVN_i}(x_i)$.

The resulting DAVN approximation can be used in two basic control strategies. Most often, the control is a booking limit control on the virtual classes. That is, a request for product j is converted into a request for the corresponding virtual class at each resource i required by product j. (Note the virtual class on each of these resources need not be the same.) If the virtual class on each resource is available, the request is accepted. If the virtual class on one or more resources is closed, the request is rejected. Thus, once the indexing from products to virtual classes is performed, the control logic is an independent, nested allocation class-level control at each resource in the network. This is the primary appeal of - and motivation for - the method in the airline industry, because it produces the sort of booking-class-level controls that are widely used by CRSs.

However, DAVN can also be used to produce bid-price controls. The bid price for resource i is simply given by

$$\pi_i^{DAVN}(t, \mathbf{x}) = \Delta V_t^{DAVN_i}(x_i)$$

where as usual $\Delta V_t^{DAVN_i}(x_i) = V_t^{DAVN_i}(x_i) - V_t^{DAVN_i}(x_i - 1)$ denotes the marginal value generated by model i.

Regardless of the control method, typically the network model that was used to generate the static bid prices $\bar{\pi}$ is re-solved and the products are re-indexed periodically as demand conditions change. In the airline industry, for example, the indexing process is a fairly major change to the CRS, so often it is only done on a seasonal basis.

Simulation-based optimization to compute optimal control parameters: Consider optimizing nested protection levels of a virtual nesting scheme. The idea is to first fix an indexing scheme and a nesting order on each resource (for example, using DAVN), and then to set nested protection levels (or booking limits) for each resource based on this nesting order. Using a network-level simulation, one can generate samples of demand and compute stochastic gradients - "noisy" estimates of the partial derivatives of the network revenue with respect to the protection level parameters of the control policy. This gradient information can then be used in a steepest descent algorithm to search for a network-optimal (rather than resource-level optimal) set of protection level parameters. van Ryzin and Vulcano (2006) propose such a stochastic gradient method assuming continuous demand and continuous capacity. They show how the stochastic gradients can be efficiently computed.

Bertsimas and de Boer (2005) focus on the discrete demand, discrete capacity problem, and their proposed methods relies on first difference rather than first gradient estimates. We refer the reader to these two articles for more information.

4 CRITICAL ISSUES: DEMAND MODELS, OVERBOOKING, MEASURING EFFECTIVENESS OF RM

Note that the models we introduced above made specific, possibly unrealistic, assumptions about the demand; see our discussion in Section 2.2. Recent line of research relaxes the simplistic demand assumptions of the traditional models, trying to capture explicitly the choice behavior that customers display when faced with multiple alternatives (products or fare classes) in a purchase context. Talluri and van Ryzin (2004b) develop the choice-based RM theory for the single-leg case, and van Ryzin and Vulcano (2008) propose a simulation-based optimization approach that improves an initial set of network virtual protection levels accounting for choice behavior effects. Demand modeling remains a critical decision in all RM applications, with implications for both forecasting and optimization.

Note that all of the models introduced so far assume there are no cancelations or no-shows, which, in fact, are an integral part of the RM problem in travel and hospitality industries. Cancelations bring an additional layer of complexity, especially to the network RM problem. Typically, overbooking and booking control decisions are carried out separately in airlines, and *virtual capacities* computed in the overbooking module of a RM system is the main input to booking control. This reduces the complexity of forecasting and optimization significantly. We refer the reader to Talluri and van Ryzin (2004a) and Karaesmen and van Ryzin (2008) for more information. For use of simulation-based optimization in solving overbooking problems, we refer the reader to Karaesmen and van Ryzin (2004).

In addition to development of methodology, one critical issue in RM is measuring effectiveness of RM systems. Simulation is used to measure the revenue opportunity during pre-implementation phase and to measure revenue benefits post-implementation. This latter topic and other issues are covered in more depth in Talluri and van Ryzin (2004a), along with a broader range of RM-related problems on dynamic pricing and auctions that are important in other business contexts.

5 CONCLUSIONS

RM is now a highly developed scientific and professional practice in the airline industry. This airline success has lead to a rapidly growing interest in using RM techniques in other industries. We have focused our attention on the core methodology developed for use in the airline industry (and related industries like hotels) over the last 25 years. We also tried to give a glimpse of use of simulation in RM.

Current industry adopters of RM include hotels, car rental companies, shipping companies, television and radio broadcasters, energy transmission companies, manufacturing, advertising, financial services, and apparel retailers. With each new industry application, one encounters new challenges in both modeling, forecasting and optimization, so research in the area continues to grow. While the details of RM problems can change significantly from one industry to the next, the focus is always on making better demand decisions - and not manually with guess work and intuition - but rather scientifically with models and technology, all implemented with disciplined processes and systems.

ACKNOWLEDGMENTS

This paper originally appeared in the Proceedings of the Winter Simulation Conference in 2008. We thank the conference organizers for inviting us to resubmit the paper and to contribute to the Advanced Tutorials in WSC'09.

REFERENCES

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

Belobaba, p. 2001. Revenue and competitive impacts of O-D control: Summary of PODS results. In *First Annual INFORMS Revenue Management Section Conference,* Columbia University, New York, NY, 2001.

Bertsimas, D. and S. de Boer. 2005. Simulation-based booking limits for airline revenue management. *Operations Research,* 53(1):90–106.

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

Huh, W.T. and P. Rusmevichientong. 2007. Online sequential optimization with biased gradients: Theory and applications to censored demand. Working paper, Columbia University, New York, NY.

Karaesmen, I. and G. van Ryzin. 2004. Overbooking with substitutable inventory classes. *Operations Research,* 52(1): 83–104.

Karaesmen, I. and G. van Ryzin. 2008. A Practical Approach to Overbooking in Network Revenue Management. Working paper, University of Maryland, College Park, MD.

Kunnumkal, S. and H. Topaloglu. 2008. A stochastic approximation method for the single-leg revenue management problem with discrete demand distributions. Working paper, Cornell University, Ithaca, NY.

Lan, Y., H. Gao, M. Ball and I. Karaesmen. 2007. Revenue management with limited information. Forthcoming in *Management Science*.

Littlewood, K. 1972. Forecasting and control of passenger bookings. In *Proceedings of the Twelfth Annual AGIFORS Symposium,* Nathanya, Israel, 1972.

Robinson, L. 1995. Optimal and approximate control policies for airline booking with sequential nonmonotonic fare classes. *Operations Research,* 43(2): 252–263.

Smith, B., J. Leimkuhler and R. Darrow. 1992. Yield Management at American Airlines. *Interfaces,* 22(1):8–31.

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

Talluri, K. and G. van Ryzin. 1999. A randomized linear programming method for computing network bid prices. *Transportation Science,* 33(2):207–216.

Talluri, K. and G. van Ryzin. 2004a. *The Theory and Practice of Revenue Management.* Kluwer Academic Publishers, Norwell, Massachusettes.

Talluri, K. and G. van Ryzin. 2004b. Revenue management under a general discrete choice model of consumer behavior. *Management Science,* 50(1):15–33.

van Ryzin, G. and J. McGill. 2000. Revenue management without forecasting or optimization: An adaptive algorithm for determining airline seat protection levels. *Management Science,* 46(6): 760–775.

van Ryzin, G. and G. Vulcano. 2006. Simulation-based optimization of virtual nesting controls for network revenue management. Forthcoming in *Operations Research.*

van Ryzin, G. and G. Vulcano. 2008. Computing virtual nesting controls for network revenue management under customer choice behavior. *M&SOM,* 10(3):448–467.

Williamson, E. 1992. *Airline Network Seat Inventory Control: Methodologies and Revenue Impact.* PhD thesis, MIT, Cambridge, MA.

AUTHOR BIOGRAPHIES

KALYAN T. TALLURI is ICREA researcher and faculty member in the Department of Business and Economics at University of Pompeu Fabra in Barcelona, Spain. He has a PhD in Operations Research from MIT. His main research interest is in revenue management. He has won numerous awards for his scientific work in this field, including the INFORMS Lancaster Prize in 2005 for the book *The Theory and Practice of Revenue Management,* (Kluwer Academic Publishers, 2004). His email address is <kalyan.talluri@upf.edu>.

GARRETT J. VAN RYZIN is Paul M. Montrone Professor of Private Enterprise in Decision, Risk and Operations Department at the Columbia University Graduate School of Business. He received his Ph.D. in Operations Research from MIT. His research interests include stochastic optimization, pricing and revenue management and supply chain management. He has won numerous awards for his scientific work, including the INFORMS Lancaster Prize in 2005 for the book *The Theory and Practice of Revenue Management,* (Kluwer Academic Publishers, 2004). His email address is <gjv1@columbia.edu>.

ITIR Z. KARAESMEN is an Assistant Professor of Operations Management in the Robert H. Smith School of Business at University of Maryland, College Park. She has a PhD in Management Science from Columbia University. Her research interests are in revenue management and inventory management of perishables. Her email address is <ikaraes@rhsmith.umd.edu>.

GUSTAVO J. VULCANO is an Assistant Professor of Operations Management at the Stern School of Business, New York University. He has a PhD in Decision, Risk and Operations from Columbia University. His primary research areas include the development of models and computational methods for revenue management. His email address is <gvulcano@stern.nyu.edu >.