Operations Research

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Analysis of Deterministic LP-Based Booking Limit and Bid Price Controls for Revenue Management

Stefanus Jasin, Sunil Kumar

OPERATIONS RESEARCH
Vol. 61, No. 6, November–December 2013, pp. 1312–1320
ISSN 0030-364X (print) | ISSN 1526-5463 (online)

http://dx.doi.org/10.1287/opre.2013.1216

Analysis of Deterministic LP-Based Booking Limit and Bid Price Controls for Revenue Management

Stefanus Jasin
Stephen M. Ross School of Business, University of Michigan, Ann Arbor, Michigan 48109, sjasin@umich.edu

Sunil Kumar
Booth School of Business, University of Chicago, Chicago, Illinois 60637, sunil.kumar@chicagobooth.edu

We study the performance of two popular and widely used heuristics for revenue management known as the booking limit and bid price controls. In contrast to a recent result in the literature where frequent re-solvings of a certain heuristic are shown to significantly reduce revenue loss, we show that the asymptotic revenue loss of either booking limit or bid price control *cannot* be reduced regardless of the choice of re-solving times and the frequency of re-solving. Moreover, we also show that further variations within the policy classes, such as *nested* instead of *partition* booking limit, or *certainty equivalent* instead of *additive* bid price, are simply indistinguishable in terms of their order of revenue loss under frequent re-solvings. This negative result highlights the limitation of re-solving deterministic linear programs when the solution is interpreted as either a booking limit or a bid price. Finally, we briefly discuss how to modify the traditional booking limit control to make it more responsive to frequent re-solvings and test its performance using numerical experiments.

Subject classifications: inventory/production; approximations/heuristics; revenue management; reoptimization; asymptotic optimality.
Area of review: Stochastic Models.
History: Received October 2010; revisions received December 2011, September 2012; accepted August 2013. Published online in *Articles in Advance* December 12, 2013.

1. Introduction

We consider a standard revenue management (RM) setting in which a firm is faced with the problem of allocating a limited amount of resources over a finite horizon. The resources are consumed by products that are sold at fixed prices and demand for each product arrives according to a time-homogeneous Poisson process. The objective is to design a dynamic control policy that maximizes expected total revenue by allocating resources to products appropriately. Given the computational intractability of the problem as stated above, the literature is replete with heuristics solutions. Among these is a class of heuristics that use the solution of the so-called deterministic linear program (DLP) that results when all random variables in the problem are replaced by their expected values, possibly conditioned on the observed state of the system. Heuristics within this class differ in two ways. First, they differ in how they incorporate state information. Second, they differ in how they interpret the solution of the DLP as control parameters for some implementation. Common implementations include booking limit and bid price controls (see Talluri and van Ryzin 2004 chapter 3). There are several popular variants of booking limit and bid price controls. (We discuss this in detail in §3.) In their basic form, DLP-based heuristics solve the DLP just once, at the beginning of the selling horizon. They interpret the DLP solution as yielding parameters for a control policy and then implement this policy *statically* throughout the horizon. So, they not only ignore stochastic variability in demand, but also fail to correct for the error as the variability is revealed. Despite this, it is known that when the scale of the problem as denoted by k becomes large, the expected revenue loss when using a static DLP-based heuristic that interprets the solution as booking limit is no worse than $O(\sqrt{k})$ (Cooper 2002). Similarly, it is known that a static DLP-based heuristic that interprets the solution as bid price has an expected revenue loss of $O(\sqrt{k})$ (Talluri and van Ryzin 1998). (To get $O(\sqrt{k})$, an assumption that a sale results in random revenue that is *continuously* distributed is needed. This assumption does not hold in our *discrete price* setting, see §2.)

A case can be made that any reduction in asymptotic loss $O(\sqrt{k})$ is highly desirable especially in applications like airline revenue management. The $O(\sqrt{k})$ loss can be attributed to three potential causes. First, the asymptotic loss is calculated against an unattainable upper bound on revenue since the optimal revenue is hard to calculate in general. It is conceivable that this bound is too loose. Indeed, this is the case in the setting of Brumelle and McGill (1993), where even the optimal control has $O(\sqrt{k})$ loss when compared to the bound. In our setting, this is not the case. Second, the loss could be due to static implementation that could be potentially improved by incorporating the updated state information through re-solving. The impact of re-solving has been intensively studied in the literature. Cooper (2002) was the first to show that re-solving the DLP and implementing the solution as booking limits may actually backfire. In various settings different from ours, Maglaras and Meissner (2006) and Chen and Homem-de-Mello (2010) established that re-solving cannot have lower expected revenue than a static scheme. In the same spirit, Secomandi (2008) also provided sufficient conditions that guarantee re-solving will not result in lower expected revenue than a static scheme. Third, the problem could be due to the "wrong" use of the DLP solution to construct the heuristic implementation in the first place, which no amount of re-solving can remedy. To the best of our knowledge, Reiman and Wang (2008) were the first to show that re-solving the DLP, combined with an appropriate implementation, can significantly reduce asymptotic revenue loss. Interpreting the DLP solution as admission probabilities (see §6 for a definition), they proved that a *single* re-solve at a properly chosen time is sufficient to reduce the expected revenue loss from $O(\sqrt{k})$ to $o(\sqrt{k})$. In a recent predecessor paper (Jasin and Kumar 2012), we showed that the revenue loss of this probabilistic control combined with an appropriate choice of re-solving schedule can, in fact, be bounded by a constant *independent* of the scale of the problem. That is, the expected revenue loss is $O(1)$ under sufficiently frequent re-solvings. This establishes an important fact that a DLP-based heuristic can indeed have an excellent performance, which is asymptotically indistinguishable from the optimal policy.

A natural question is whether the $O(1)$ asymptotic loss can also be obtained by using either the booking limit or bid price interpretation of the DLP, perhaps when combined with an appropriate re-solving schedule. This paper addresses this very question and answers it in the *negative*. We show that, regardless of frequency, re-solving *cannot* reduce the expected revenue loss of $O(\sqrt{k})$ when the DLP solution is implemented via a traditional booking limit or bid price. We also establish that, with sufficiently frequent re-solvings, further variations within the policy classes, such as *nested* instead of *partition* booking limits, or *certainty equivalent* instead of *additive* bid prices, are simply *indistinguishable* in their order of revenue loss. To be precise, the revenue difference between any two of these variants is $O(1)$. Because of the tractability of the analysis, we only prove these results for the setting of single-leg RM (i.e., multiple products sharing one unique resource, as in the case of a single flight with multiple economy fare classes). However, we conjecture that the same results also hold in general network RM settings. We test this using numerical experiments in §7.

The remainder of this paper is organized as follows: in §2, we lay out the model and the assumptions used in the paper, the performance measure of interest, namely, the expected revenue loss or regret, and the asymptotic setting, which formalizes the notion of *scale* or relative *size*. (We use revenue loss and regret interchangeably.) In §3, we define the various heuristics that will be analytically studied in the paper. The probabilistic control of Reiman and Wang (2008) and Jasin and Kumar (2012) will be briefly discussed in §6. Sections 4 and 5 analyze the asymptotic performance of booking limit and bid price controls, respectively. In §6, we introduce an improvement of booking limit. Finally, in §7, we provide numerical results and, in §8, we conclude the paper by addressing some limitations as well as potential future research directions. Unless otherwise noted, all the proofs can be found in the electronic companion of this paper (available as supplemental material at http://dx.doi.org/10.1287/opre.2013.1216).

2. Model, Setting, and Assumptions

2.1. Model

We consider a system with n products (indexed by $j = 1, \ldots, n$) and one resource. This setting is often called *single-leg* RM in the literature. (It owes its name to the early application of RM in the airline industry.) We assume that product j is sold at a fixed price f_j and that our resource has initial capacity C. Let f denote the price vector (unless otherwise noted, all vectors are to be understood as column vectors). The sale of one unit of product j consumes $a_j > 0$ units of resource and requests for that product arrive according to an independent Poisson process with rate λ_j, with $\lambda := (\lambda_j) \geqslant 0$. (Extension to time-inhomogeneous Poisson arrival is straightforward but we do not attempt that here.) Let $\Lambda(s,t) = (\Lambda_j(s,t))$ denote the number of arriving requests during time interval $[s,t)$, $0 \leqslant s < t \leqslant 1$. Also, let $Z_\pi(s,t) = (Z_{\pi,j}(s,t))$ denote the number of accepted requests (sold products) under control π during $[s,t)$.

2.2. The Stochastic and Deterministic Problems

The objective of maximizing total expected revenue earned over the selling horizon results in the following stochastic optimization problem:

$$\begin{aligned} J_{\text{opt}} = \sup_{\pi \in \Pi} \; & \mathbf{E}\left[\sum_{j=1}^{n} f_j Z_{\pi,j}(0,1)\right] \\ \text{s.t.} \; & \sum_{j=1}^{n} a_j Z_{\pi,j}(0,1) \leqslant C \quad \text{and} \\ & 0 \leqslant Z_\pi(0,t) \leqslant \Lambda(0,t) \quad \forall\, t \in [0,1), \end{aligned} \tag{1}$$

where the constraints must be satisfied with probability 1 and Π is the set of admissible policies (i.e., the set of all policies that determine whether to accept an arriving request at time t based only on the available information acquired up to time t, including the type of the arriving request). The standard deterministic analog of the problem above is obtained by replacing random quantities in (1) with their expected values. This results in the following linear program:

$$\textbf{pDLP}[C,\lambda]: \quad \text{maximize} \quad \sum_j f_j y_j$$
$$\text{s.t.} \quad \sum_j a_j y_j \leqslant C \quad \text{and} \qquad (2)$$
$$0 \leqslant y \leqslant \lambda.$$

We call (2) the primal deterministic linear program (pDLP, or simply DLP) and denote its optimal value by J_{DLP}. The dual of (2), which is useful in the discussion of bid price, is

$$\textbf{dDLP}[C,\lambda]: \quad \text{minimize} \quad Cv + \sum_j \lambda_j u_j$$
$$\text{s.t.} \quad a'v + u \geqslant f \quad \text{and} \quad u, v \geqslant 0. \qquad (3)$$

In this paper, we make the following modeling assumptions: (1) $f_j/a_j > f_{j+1}/a_{j+1}$ for all $j = 1, 2, \ldots, n-1$ and (2) $0 < C - \sum_{j=1}^{n-1} a_j\lambda_j < a_n\lambda_n$. Assumption 1 ranks the products in the order of decreasing profitability and Assumption 2 simply says that total requested consumptions exceed initial capacity and *all* products matter. (If $C - \sum_{j=1}^{\theta-1} a_j\lambda_j < a_\theta\lambda_\theta$ for some $\theta < n$, then the problem is effectively reducible to single-leg RM with θ products.) Mathematically, the later is also equivalent to saying that the original LP is *nondegenerate*. The impact of degeneracy will be briefly discussed in §8.

2.3. Re-Solving

Let Y and $[V, U]$ denote the *unique* optimal solution to $\textbf{pDLP}[C,\lambda]$ and $\textbf{dDLP}[C,\lambda]$, respectively. At any time $t \in [0, 1]$, let $C_\pi(t)$ denote the remaining capacity under control π, excluding any consequences of any arrival at t. Also, let $Y_\pi(t)$ and $[V_\pi(t), U_\pi(t)]$ denote the optimal solution to $\textbf{pDLP}[C_\pi(t), (1-t)\lambda]$ and $\textbf{dDLP}[C_\pi(t), (1-t)\lambda]$, respectively. (Note that $(1-t)\lambda$ is the expected demand during time interval $[t, 1]$.) For completeness, we define $C_\pi(0) = C$, $Y_\pi(0) = Y$, and $[V_\pi(0), U_\pi(0)] = [V, U]$. Let t_l denote the lth re-solving time and $\Gamma = \{t_l\colon 0 < t_1 < \cdots < t_N < 1\}$ denote the set of re-solving times (re-solving schedule), where N is the number of re-solvings. For notational convenience, we will use $t_0 = 0$ and $t_{N+1} = 1$. The general algorithm for a control π based on re-solving the DLP is given below.

Algorithm 1. (Re-solving control) π
Given $\Gamma = \{t_l\}$, do
Step 1. Start with $l = 0$.
Step 2. At time t_l use $[Y_\pi(t_l), V_\pi(t), U_\pi(t)]$ to construct an allocation control $\pi(t_l)$.
Step 3. Implement $\pi(t_l)$ during time interval $[t_l, t_{l+1})$ subject to capacity.
Step 4. Set $l = l+1$ and go back to Step 2 until either $l = N+1$ or capacity is zero.

Two remarks are in order. First, it is possible to devise a schedule that includes re-solving at *random* times (e.g., a sequence of hitting times). In this paper, t_l will always be used to denote a fixed time and τ_l will be used to denote a random time. Second, although the dual solution $[V_\pi(t), U_\pi(t)]$ is *unique* at time 0, it may *not* be unique at all times t. This is so because $\textbf{pDLP}[C_\pi(t), (1-t)\lambda]$ may *not* be nondegenerate at all times t. As will be discussed in §5, this creates an issue of the *nonuniqueness* of the dual. For now, we simply assume that the controller is free to pick any feasible optimal dual solution to use.

2.4. Expected Regret

Let $R_{\pi,\Gamma}(s, t)$ denote total revenue earned under control π, using re-solving schedule Γ, and during time interval $[s, t)$, $0 \leqslant s < t \leqslant 1$. For notational brevity, whenever it is clear from the context that Γ is being used, we will simply write $R_\pi = R_\pi(0, 1)$ instead of $R_{\pi,\Gamma}(0, 1)$. We use *expected regret*, defined as the expected difference between the revenue earned under optimal allocation policy and the heuristic, as performance measure. Given a sample path of request arrivals, the optimal allocation policy can be computed in *hindsight* by solving the following:

$$\textbf{Hindsight LP}: \quad \text{maximize} \quad \sum_j f_j x_j$$
$$\text{s.t.} \quad \sum_j a_j x_j \leqslant C \quad \text{and} \qquad (4)$$
$$0 \leqslant x \leqslant \Lambda(0, 1).$$

The above LP is what the controller would use if all requests that arrived could be queued and the allocations done only at the end of the horizon (x_j requests of each type j would be accepted). Naturally, this allocation rule is often called the hindsight policy. Let R_H denote the optimal value of the hindsight LP. Our definition of expected regret of π is given by

$$\text{Expected Regret} = \mathbf{E}[R_H - R_\pi].$$

The following result is straightforward. We omit its proof.

Theorem 1. $E[R_\pi] \leqslant E[R_H] \leqslant J_{DLP}$.

2.5. The Asymptotic Setting

As in Cooper (2002), Reiman and Wang (2008), and Jasin and Kumar (2012), in this paper we consider a sequence of problems with capacity $C^k = kC$ and arrival rates $\lambda^k = k\lambda$, $k > 0$. The superscript k can be interpreted as the *scale*, or relative *size*, of the problem. (If C is normalized to 1, then one can directly interpret k as the size of initial capacity.) The idea is to simultaneously increase the size of demand and capacity while maintaining their relative proportion constant. Since revenue loss tends to grow as the size of the problem increases (due to increased volatility in demand realization), this setting allows us to study the growth rate of loss as a function of the size of the problem. We call the problem with parameters $[C^k, \lambda^k]$ the kth problem. Aside from the price vector f and consumption vector a, which do not depend on k, we put superscript k on all previously defined notations as a reference to the kth problem. For example, total arrivals during time interval $[s, t), 0 \leqslant s < t \leqslant 1$, in the kth problem is denoted by $\Lambda^k(s, t)$. Similarly, we also write t_l^k and Γ^k (we allow the choice of re-solving times to depend on k), J_{opt}^k, J_{DLP}^k, and R_π^k. Given $g(k)$ and $h(k)$, the expressions $g(k) = O(h(k))$ and $g(k) = \Omega(h(k))$ mean there exists positive constants M_1 and M_2 such that $g(k) \leqslant M_1 h(k)$ and $g(k) \geqslant M_2 h(k)$ for all $k > 0$, respectively. The little-OH notation $g(k) = o(h(k))$ means $\lim_{k\to\infty} g(k)/h(k) = 0$.

3. Control Definitions

In this section we give the definition of two well-known families of DLP-based heuristics: booking limits and bid prices. Definitions of other relevant heuristics will be provided as needed. For simplicity, whenever there is no loss in meaning, we will suppress the superscript k on all notations.

3.1. Booking Limits

The basic idea of booking limits is to interpret Y as the maximum number of accepted requests (or sales) for each product. It prescribes that we continue to accept requests for product j as long as its cumulative sales has not exceeded Y_j. This simple heuristic is often called *partition booking limit control* (PBLC). The drawback of PBLC, though, is obvious: it quickly rejects requests for more profitable products when their cumulative sales have already achieved the assigned limits. Hence, a different implementation, called *nested booking limit control* (NBLC), is usually used in practice. There are *many* different forms of NBLC. To define a general booking limit, which encompasses both PBLC and NBLC, following Perakis and Roels (2010), we start with a collection $\mathbf{S}$ of the set of products $S = \{1, 2, \ldots, n\}$. We call $\mathbf{S}$ a *nesting set*. In addition, we also define a sequence of positive weighting vectors $\{\alpha_S\}_{S\in\mathbf{S}}$. (In asymptotic setting, the α_S's do not scale with k.) Finally, for each j, we introduce a new variable z_j, which should be interpreted as the cumulative sales for product j. The form of booking limits corresponding to $\mathbf{S}$ and $\{\alpha_S\}_{S\in\mathbf{S}}$ is then characterized by the set of the following inequalities:

$$\sum_{j\in S} \alpha_{S,j} z_j \leqslant \sum_{j\in S} \alpha_{S,j} Y_j \quad \text{for all } S \in \mathbf{S}.$$

These can be interpreted as follows: the weighted sum of cumulative sales of products $\{j \in S\}$ cannot exceed $\sum_{j\in S} \alpha_{S,j} Y_j$. It is easy to see that PBLC is equivalent to booking limits with $\mathbf{S} = \cup_j \{j\}$ and $\alpha_{\{j\}} = 1$. As another example, consider a single-leg RM with two products and $a_1 = a_2 = 1$. If we use $\mathbf{S} = \{\{1, 2\}, \{2\}\}$ with $\alpha_{\{1,2\}} = [1, 1]$ and $\alpha_{\{2\}} = 1$, which translates to $z_1 + z_2 \leqslant Y_1 + Y_2 = C$ and $z_2 \leqslant Y_2$, then we have the standard nesting procedure: accept at most Y_2 requests for product 2 and all requests for product 1, subject to capacity.

The above definition can be extended to *dynamic* booking limits, where we allow the controller an extra flexibility for changing a nesting set and its weighting vectors at every re-solving time. Fix $\Gamma = \{t_l\}$ and let $\mathbf{S}_l$ denote the nesting set to be applied during $[t_l, t_{l+1})$. (For brevity, we assume that the choice of $\{\alpha_S^l\}_{S\in\mathbf{S}_l}$ is already encoded within the choice of $\mathbf{S}_l$.) Let $z_{j,l}$ denote the variable to be interpreted as the cumulative sales of products j from time t_l onward. The booking limit control corresponding to $\{t_l\}$ and $\{\mathbf{S}_l\}$ is given below.

Booking limit control (BLC)
Given $\{t_l\}$ and $\{\mathbf{S}_l\}$, do
Step 1. Start with $l = 0$.
Step 2. During $[t_l, t_{l+1})$ apply allocation rules characterized by the set of inequalities

$$\sum_{j\in S} \alpha_{S,j}^l z_{j,l} \leqslant \sum_{j\in S_l} \alpha_{S,j}^l Y_j(t_l) \quad \text{for all } S \in \mathbf{S}_l$$

subject to capacity.
Step 3. Set $l = l+1$ and go back to Step 2 until either $l = N+1$ or capacity is zero.

The policy in Step 2 can be read as follows: the weighted sum of cumulative sales of products $\{j \in S\}$ during $[t_l, t_{l+1})$ cannot exceed $\sum_{j\in S} \alpha_{S,j}^l Y_j(t_l)$. Note that the choice of $\mathbf{S}_l$ could either be fixed from time 0 or adapted to the information accumulated up to time t_l. We will use $R_{BLC,\{\mathbf{S}_l\}}$ to denote total revenue earned by applying $\{\mathbf{S}_l\}$.

3.2. Bid Prices

Another well-known DLP-based heuristic is the so-called *additive bid price control* (ABPC). The basic idea is to interpret an appropriate linear combination of the dual solution as an approximate opportunity cost of selling an additional product and use it as a threshold for admission. (In single-leg RM, we only have one dual variable so the additive structure does not show up.) We give the formal definition below.

Additive bid price control
Given $\Gamma = \{t_l\}$, do
Step 1. Start with $l = 0$.
Step 2. During $[t_l, t_{l+1})$ do
Accept a request for product j if $V(t_l) a_j \leqslant f_j$ subject to capacity.
Step 3. Set $l = l+1$ and go back to Step 2 until either $l = N+1$ or capacity is zero.

The readers might notice that the above definition of ABPC is *not* well defined. In particular, the dual variable $V(t_l)$, which forms the solution to $\mathbf{dDLP}[C(t_l), (1 - t_l)\lambda]$, may *not* be unique (see §5). Since different dual solutions may lead to different allocation rules, one expects that nonuniqueness can potentially create a serious problem. Indeed, partly motivated by this, Bertsimas and Popescu (2003) introduced a different DLP-based bid price called *certainty equivalent bid price control* (CEBPC). Their idea is to use a different (and well-defined) approximation of opportunity cost computed directly from the optimal value of DLP itself. Let $J_{DLP}(t, C)$ denote the optimal value of $\mathbf{pDLP}[C, (1-t)\lambda]$. We define the resulting control below.

Certainty equivalent bid price control
Given $\Gamma = \{t_l\}$, do
Step 1. Start with $l = 0$.
Step 2. During $[t_l, t_{l+1})$ do
Accept a request for product j if $J_{DLP}(t_l, C(t_l)) - J_{DLP}(t_l, C(t_l) - a_j) \leqslant f_j$ subject to capacity.
Step 3. Set $l = l + 1$ and go back to Step 2 until either $l = N + 1$ or capacity is zero.

4. Analysis of Booking Limit Controls

We now analyze the performance of booking limits. It is known that PBLC without re-solving has $O(\sqrt{k})$ regret (Cooper 2002). The first issue to address is whether we can do better with frequent re-solvings and dynamic nesting. We resolve this in the negative, as far as asymptotic regret is concerned, in the theorem below.

Theorem 2. *There exists a positive constant M independent of $k \geqslant 1$ and the choice of re-solving schedule $\Gamma^k = \{t_l^k\}$ such that for all sufficiently large k we have*

$$\inf_{\{\mathbf{S}_l^k\}} \mathbf{E}[R_H^k - R_{BLC,\{\mathbf{S}_l^k\}}^k] \geqslant M\sqrt{k}.$$

In comparison to the $O(1)$ regret of probabilistic control (Jasin and Kumar 2012), Theorem 2 tells us that, even with dynamic nesting and frequent re-solvings, we still suffer a revenue loss of order $\Omega(\sqrt{k})$. That is, re-solving, regardless of frequency, cannot really improve on the static scheme in terms of the order of regret. (Reiman and Wang 2008 showed that, in a single-leg RM with two products, it is possible to reduce $O(\sqrt{k})$ regret to $o(\sqrt{k})$ with static booking limit (see their Equation 32). To get this result, they need to assume that demand for the first product is deterministic. This exposes the nontrivial role of randomness in the premise of Theorem 2.) Although negative news, this does not mean that nesting cannot improve over the original PBLC; indeed, an appropriate nesting improves revenue by an order of $\Omega(\sqrt{k})$ (see electronic companion EC 5). It does, however, suggest a fundamental limitation in working with booking limits, one that cannot simply be addressed by frequent re-solvings or by choosing a clever nesting scheme. We want to stress that this result does *not* change even if we allow random re-solving times.

The next issue to address is nesting, whether it is *always* useful, and if so, what is the magnitude of the revenue improvement? At the outset, we suspect that not all nestings will perform equally well. Indeed, one choice of a nesting set can significantly outperform another by an order of $\Omega(k)$. This motivates the division of nesting sets into two groups: the *good* (type G) and the *dangerous* (type D). Define three sets of indices as follows: $J_\lambda(t) = \{j\colon Y_j(t) = (1-t)\lambda_j\}$, $J_y(t) = \{j\colon 0 < Y_j(t) < (1-t)\lambda_j\}$, and $J_0(t) = \{j\colon Y_j(t) = 0\}$. Product $j \in J_\lambda(t)$ is said to be *full* and product $j \in J_y(t)$ is said to be *fractional*. We give a definition.

Definition 1. A nesting set $\mathbf{S}$ is said to be *good* (of type G) at time t if the *singleton* $\{j\} \in \mathbf{S}$ for all $j \in J_y(t)$ and *dangerous* (of type D) otherwise.

Intuitively, since the motivation for nesting is to give priority to more profitable products, following the structure of the LP solution, we propose that "good" nestings should *not* give priority to fractional products when it comes to resource allocation. (Obviously, PBLC is of type G.) Let $\Pi_G(t)$ and $\Pi_D(t)$ denote the set of good and dangerous nesting sets with respect to the solution of DLP at time t, respectively. (Per our definition above, a nesting set that is good at time t may not be good at time $t' \neq t$.) Also, let $\tau_{i\theta}^k$ denote the arrival time of the $(i\theta)$th customer, where $\theta \in \mathbf{Z}^+$. We state a theorem.

Theorem 3. *Suppose that we simultaneously re-solve both BLC and PBLC at times $\{\tau_{i\theta}^k\}_{i\geqslant 1}$. If we always choose good nesting set $\mathbf{S}_i^k \in \Pi_G(\tau_{i\theta}^k)$ for all i, then*

$$\mathbf{E}[|R_{BLC,\{\mathbf{S}_i^k\}}^k - R_{PBLC}^k|] \leqslant \theta\left[\sum_j f_j\lambda_j\right].$$

(If we set $\theta = 1$, then we are in effect re-solving for every incoming customer.) Theorem 3 offers an important operational insight. Although, as previously discussed, nesting can be advantageous, its benefit *decreases as the frequency of re-solving increases*. In particular, as an immediate corollary of Theorem 3, under sufficiently frequent re-solvings, *all* good BLC (i.e., BLC with good nesting sets) are within $O(1)$ of one another. This is true regardless of the good nesting sets being used in the implementation. So, they all have the same limiting performance. This has an interesting practical implication: if we plan to re-solve the DLP *sufficiently frequently*, then we might as well use the standard PBLC instead of trying to figure out the "best" nesting. (Although the implementation of NBLC is sometimes easier, the task of finding the best nesting may not be an easy one, especially in general network settings.)

Remark 1. Since the set $\Pi_G(\cdot)$ may change from time to time, the above result may *not* hold if, for example, we use the same nesting set $\mathbf{S}_i = \mathbf{S} \in \Pi_G(0)$ for all i (at all times $\{\tau_{i\theta}\}$). This is because although $\mathbf{S}$ is good at time 0, it may not be good at any other re-solving times. As an illustration, consider a single-leg RM with three products and $a_1 = a_2 = a_3 = 1$. Suppose that $J_\lambda(0) = \{1, 2\}$ and $J_y(0) = \{3\}$. We consider two choices of nesting sets

$\mathbf{S}^1 = \{\{1,2,3\},\{2,3\},\{3\}\}$ and $\mathbf{S}^2 = \{\{1,2,3\},\{1,3\},\{3\}\}$, each with weighting vectors equal to ones. (Note that $\mathbf{S}^1$ gives the highest priority to product 1 whereas $\mathbf{S}^2$ gives the highest priority to product 2.) Obviously, both $\mathbf{S}^1$ and $\mathbf{S}^2$ belong to $\Pi_G(0)$. And yet, whereas the performance gap between PBLC and BLC with $\mathbf{S}^1$ decreases to $O(1)$ as re-solving frequency increases ($\mathbf{S}^1$ is *always* good at any re-solving times), the gap between BLC with $\mathbf{S}^1$ and $\mathbf{S}^2$ actually increases and scales with k ($\mathbf{S}^2$ is not always good. See simulation results in electronic companion EC 6), which implies that the gap between PBLC and BLC with $\mathbf{S}^2$ also scales with k. The important lesson is it appears safer, at least from a theoretical perspective, to always use good nestings.

5. Analysis of Bid Price Controls

We now turn our attention to bid price. It is known that some bid price controls (Talluri and van Ryzin 1998, Bertsimas and Popescu 2003) have $O(\sqrt{k})$ regret. The first issue that we will address in this section is the performance of ABPC under a general DLP-based re-solving scheme.

Theorem 4. *There exists a positive constant M independent of $k > 0$, the choice of re-solving schedule $\Gamma^k = \{t_l^k\}$, and the choice of feasible $\{V^k(t_l^k)\}$ such that for all sufficiently large k*

$$\mathbf{E}[R_H^k - R_{ABPC}^k] \geqslant M\sqrt{k}.$$

The above theorem tells us that ABPC suffers the same problem as booking limit. Although re-solving does improve revenue with ABPC (without re-solving, ABPC has a $\Theta(k)$ regret because it accepts *all* incoming requests for *all* products subject to capacity), the asymptotic regret is still at least $\Omega(\sqrt{k})$ even with frequent re-solvings. As in the case of booking limit, this result does not change even if we allow random re-solving times.

In §3, we briefly mentioned that, although $V^k(0)$ is unique (by nondegeneracy assumption), there is no guarantee that $\{V^k(t_l^k)\}$ will also be unique for all times $\{t_l^k\}$. We want to know whether nonuniqueness is really a problem—and if so, when. Also of interest is the comparison between ABPC with CEBPC and booking limit. In relation to CEBPC, we want to know how bad ABPC actually is in comparison to CEBPC. In relation to booking limit, the often-mentioned benefit of bid price over booking limit is that it requires less memory to save the dual solution than the primal solution. However, there are a lack of theoretical results in the literature comparing the performance of the two. To resolve these questions, we will focus on the setting of a single-leg RM with only two products. We conjecture that the same results also hold for a more general network setting (see §7). We start with a definition.

Definition 2. A sequence $\mathbf{V} = \{V_{ABPC}(t_l)\}$ is called a *feasible* sequence of dual solution if $V_{ABPC}(t_l)$ corresponds to an optimal solution of $\mathbf{dDLP}[C(t_l), (1-t_l)\lambda]$.

A similar definition applies when we substitute t_l with τ_l. We denote the resulting revenue under $\mathbf{V}$ by $R_{ABPC,\mathbf{V}}$. Let $\tau_{i\theta}^k$ be defined as in the previous section. We state a theorem.

Theorem 5. *Consider a single-leg RM with $n = 2$. Suppose that we re-solve the DLP at times $\{\tau_{i\theta}^k\}_{i\geqslant 1}$. Then, there exists a positive constant M independent of $k > 0$ and θ such that*
1. $\sup_{\mathbf{V},\mathbf{V}^*} \mathbf{E}[|R_{ABPC,\mathbf{V}}^k - R_{ABPC,\mathbf{V}^*}^k|] \leqslant M\theta$,
2. $\sup_{\mathbf{V}} \mathbf{E}[|R_{ABPC,\mathbf{V}}^k - R_{CEBPC}^k|] \leqslant M\theta$,
3. $\sup_{\mathbf{V}} \mathbf{E}[|R_{ABPC,\mathbf{V}}^k - R_{PBLC}^k|] \leqslant M\theta$,

where the supremum is taken over the set of feasible sequence of dual solution.

Several comments are in order. First, it is not difficult to create an example that shows that, with *infrequent* re-solving (e.g., only a few times), different ABPC heuristics can outperform one another by an order of $\Omega(k)$. The first part of Theorem 5 tells us that, under *sufficiently frequent* re-solvings, the asymptotic losses of any pair of ABPC heuristics are within $O(1)$ of each other. So, *the impact of nonuniqueness on revenue loss decreases as the frequency of re-solving increases*. This suggests that one may not need to worry about nonuniqueness if the bid price is updated frequently enough. Second, although previous simulation studies show that CEBPC outperform ABPC (see Bertsimas and Popescu 2003), the second part of Theorem 5 tells us that, under *sufficiently frequent* re-solvings, the difference between ABPC and CEBPC is actually negligible. Finally, when it comes to ABPC and booking limit, it does seem that there is a clear trade-off between performance and computational power. Our simulation results in §7 suggest that, with infrequent re-solving, PBLC can significantly outperform ABPC. And yet, with increasingly frequent re-solvings, the performance gap between PBLC and ABPC decreases to $O(1)$. Thus, if one is equipped to re-solve the DLP sufficiently frequently, the third part of Theorem 5 provides a new justification for using bid price instead of booking limit.

6. Improving Booking Limits

As is evident in §4, booking limits do not do very well in comparison to the probabilistic control used in Reiman and Wang (2008) and Jasin and Kumar (2012). (Given the solution of the primal LP at time t_l, during time interval $[t_l, t_{l+1})$, the so-called *probabilistic allocation control* (PAC) accepts incoming request for product j with probability $Y_j(t_l)/[(1-t_l)\lambda_j]$ subject to available capacity.) The key explanatory insight is that booking limit does not "act" until it is too late in some sense. This is because the standard implementation of booking limit uses the remaining quantity interpretation of the DLP solution (i.e., it *greedily* accepts all requests for a given product as long as its cumulative sales has not exceeded the prescribed limit), rather than a rate-based interpretation of PAC (which proportionately *distributes* admissions throughout the selling horizon). One possible fix for this is to design a "distributed" version of booking limit that mimics PAC. In each interval, this distributed control attempts to admit exactly the expected number of admitted requests under PAC. The result is what we term DBLC. Below, we give the definition for two variants of DBLC: DBLC-U (up variant) and DBLC-D (down variant).

Distributed booking limit control (DBLC)

Given $\Gamma = \{t_l\}$, do

Step 1. Start with $l = 0$.

Step 2. During $[t_l, t_{l+1})$, accept a request for product $j \in J_y \cup J_0$ if

(i) $Z_j(t_l, t) + 1 \leqslant (t_{l+1} - t_l) Y_j(t_l)/(1 - t_l)$ (for DBLC-D)

(ii) $Z_j(t_l, t) + 1 \leqslant \lceil (t_{l+1} - t_l) Y_j(t_l)/(1 - t_l) \rceil$ (for DBLC-U)

subject to capacity.

Step 3. Set $l = l + 1$ and go back to Step 2 until either $l = N + 1$ or capacity is zero.

Given that the expected number of accepted requests can be small and fractional if the interval is small, the performance of DBLC can vary depending on whether we round up or round down, and this is especially salient at high re-solving frequencies. Hence, we have introduced the two variants above. We do not provide any theoretical analysis for DBLC. The next section has a numerical study that establishes that they can outperform the more standard booking limit controls.

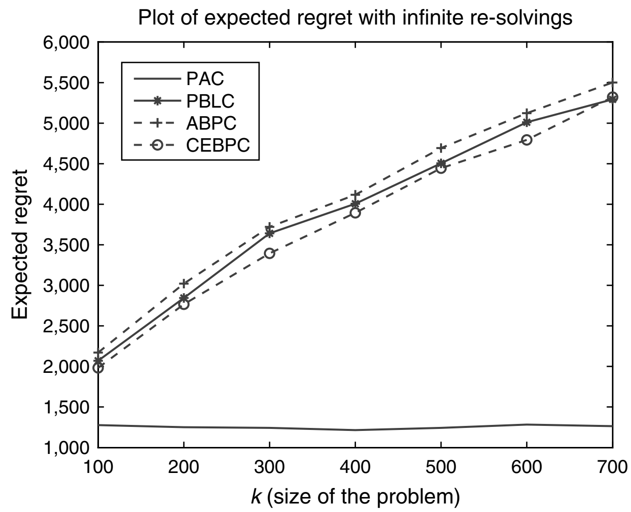
7. Numerical Studies

The purpose of this section is to illustrate via simulation studies the generality of some of the key results proved for single-leg RM in §§4 and 5, as well as to bring out a few nuances that the asymptotic analysis must necessarily miss. Our example is intended to be representative of an airline setting. We have four cities (P, Q, R, S), six flights ($F_i, i = 1, \ldots, 6$), and 10 products (10 different itineraries). F_1 represents the morning flight connecting P and Q; F_2, the morning flight connecting R and S; F_3, the noon flight connecting P and Q; F_4, the noon flight connecting city Q and city R; F_5, the noon flight connecting P city R; and F_6, the noon flight connecting R and S. The details of the 10 products are given in Table 1. We assume that only a single seat is required from each flight(s) for each product. Finally, capacity of all flights are normalized to 1, i.e., $C = [1, 1, \ldots, 1]'$, which will be appropriately scaled by k in two different scenarios.

Table 1. Details of the products used in the simulation example.

Product no.	Time	Class	Route	Price	Demand rate (λ)
1	Morning	Business	$P \to Q$	400	0.5
2	Morning	Leisure	$P \to Q$	250	1
3	Morning	Business	$R \to S$	500	0.5
4	Morning	Leisure	$R \to S$	350	1
5	Noon	Leisure	$P \to Q \to R$	200	1
6	Noon	Leisure	$P \to Q$	150	0.5
7	Noon	Leisure	$Q \to R$	100	1
8	Noon	Leisure	$P \to R$	250	0.5
9	Noon	Leisure	$P \to R \to S$	400	1
10	Noon	Leisure	$R \to S$	250	1

Figure 1. Plot of expected revenue loss with infinite re-solving.



7.1. Simulation Scenario 1

We simulate the system by varying the leg capacity, i.e., the number of seats on each flight, $k = 100, 200, \ldots, 700$. For each k we run PAC (see §6), PBLC, ABPC, and CEBPC with k re-solves (this scenario is roughly equivalent to re-solving for every incoming customer or re-solving infinitely often) and then we plot the resulting average revenue loss, when compared to the DLP upper bound, as a function of k. The plot can be seen in Figure 1 and the complete numerical results can be seen in Table 2. Some quick observations: first, as expected from Jasin and Kumar (2012), the revenue loss of PAC is seen to be bounded by a constant independent of the size of the problem. Second, the revenue loss of PBLC, ABPC, and CEBPC are increasing with k in a roughly parabolic manner. This confirms Theorems 2

Table 2. Performance with infinite re-solving.

	PAC		PBLC		ABPC		CEBPC	
k	Regret	Std	Regret	Std	Regret	Std	Regret	Std
100	1,277	27	2,071	25	2,171	27	1,982	26
200	1,250	25	2,845	34	3,021	28	2,766	35
300	1,243	24	3,641	44	3,720	47	3,395	44
400	1,215	23	4,007	51	4,116	51	3,895	52
500	1,242	25	4,505	58	4,693	57	4,445	61
600	1,283	26	5,011	61	5,123	61	4,794	61
700	1,263	24	5,294	68	5,501	70	5,321	67

Figure 2. Plot of average revenue loss under periodic re-solving.

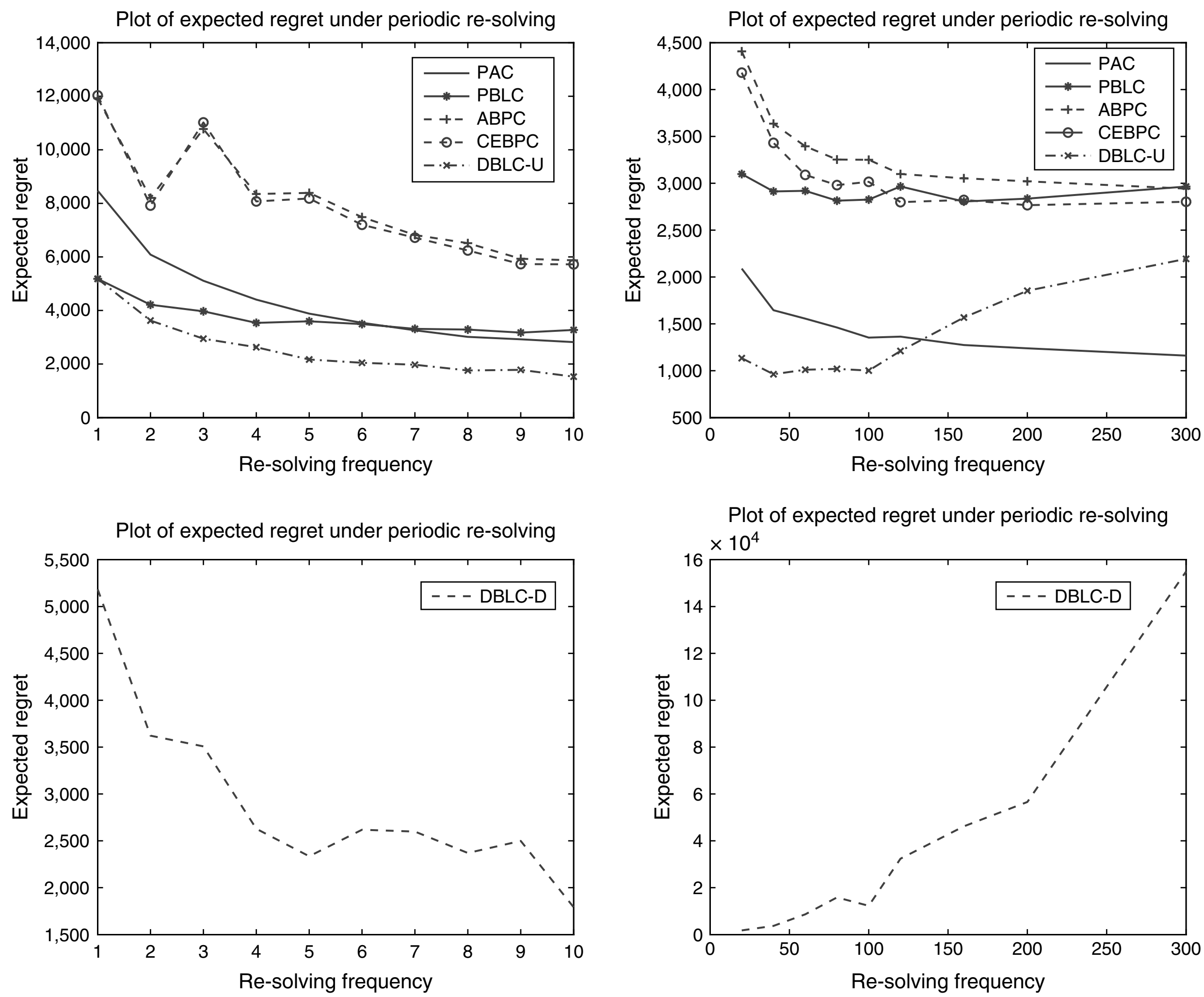


Table 3. Performance of periodic re-solving.

	PAC		PBLC		ABPC		CEBPC		DBLC-D		DBLC-U	
N	Regret	Std	Regret	Std	Regret	Std	Regret	Std	Regret	Std	Regret	Std
1	8,471	137	5,184	75	11,927	61	12,028	62	5,184	75	5,184	75
2	6,087	101	4,216	55	8,203	85	7,920	86	3,622	53	3,622	53
3	5,109	80	3,970	45	10,775	80	11,020	82	3,508	47	2,947	43
4	4,406	71	3,538	42	8,348	82	8,072	84	2,629	39	2,628	38
5	3,882	61	3,597	39	8,392	82	8,180	85	2,335	34	2,172	30
6	3,542	55	3,493	37	7,484	79	7,201	77	2,618	32	2,049	28
7	3,259	51	3,317	36	6,819	72	6,717	71	2,599	33	1,975	28
8	3,015	48	3,286	38	6,516	73	6,242	69	2,370	31	1,761	26
9	2,923	47	3,174	36	5,933	65	5,730	67	2,499	32	1,784	26
10	2,818	44	3,272	38	5,874	62	5,722	62	1,792	28	1,527	23
20	2,090	32	3,098	34	4,407	46	4,180	48	1,789	29	1,134	17
40	1,646	27	2,913	37	3,636	43	3,431	41	3,706	50	963	14
60	1,556	27	2,919	36	3,394	39	3,090	39	8,623	80	1,011	17
80	1,461	26	2,814	36	3,253	40	2,981	37	15,870	104	1,020	16
100	1,353	25	2,826	36	3,251	39	3,016	38	12,296	96	1,002	17
120	1,364	25	2,966	35	3,099	37	2,799	37	32,402	145	1,211	24
160	1,275	25	2,805	34	3,053	38	2,822	37	46,215	147	1,568	25
200	1,240	24	2,836	36	3,021	38	2,766	35	56,595	150	1,854	26
300	1,161	23	2,965	35	2,943	36	2,802	37	154,950	147	2,194	29

and 4. In addition, the revenue loss of PBLC, ABPC, and CEBPC seems to differ only by a constant order. This confirms Theorem 5.

7.2. Simulation Scenario 2

We now study the impact of re-solving frequency. To do this, we fix the system capacity at $k = 200$ and simulate the system under periodic PAC, PBLC, ABPC, CEBPC, DBLC-D, and DBLC-U, the last two were introduced in §6. The results can be seen in Figure 2 and Table 3. Some comments: first, re-solving PBLC and ABPC do result in improved performance (although the amount of improvement quickly tappers off). Of particular interest, the revenue loss of ABPC is approaching PBLC as re-solving frequency increases. This suggests the robustness of Theorem 5 in network setting. Second, the simulation shows a subtle performance difference between DBLC-D and DBLC-U. Intuitively, if the interval is small (e.g., at high re-solving frequency), the expected number of accepted requests can be small enough that rounding has the potential of introducing significant error in acceptance. (PAC does not share this issue because it does its allocation probabilistically.) To our knowledge, this is the first simulation study that shows that an otherwise negligible issue such as rounding up versus rounding down can seriously damage performance.

8. Concluding Remarks

This paper delivers bad news about the limited impact of re-solving on DLP-based booking limits and bid prices. It also delivers moderately good news that finer distinctions within the policies may not matter when re-solving is sufficiently frequent. These results are derived based on a nondegeneracy assumption on the original DLP. If the original DLP is *degenerate*, that is $C = \sum_{j=1}^{n} a_j \lambda_j$, or equivalently $J_y(0) = \varnothing$, then re-solving booking limits and bid prices can potentially reduce their asymptotic regret. It can be shown, for example, that ABPC *without* re-solving has $O(\sqrt{k})$ regret instead of $\Omega(k)$. It can also be argued that ABPC can have $o(\sqrt{k})$ regret with sufficiently frequent re-solvings. Several issues appear pertinent for future research. First, the negative results of Theorems 2 and 4 assume that the booking limits and bid prices are set by solving a DLP that uses *mean arrival rates* information. It is not clear whether the conclusions still hold if we use something other than the mean rates. Indeed, moving away from DLP to a larger class of "semi-deterministic" heuristics, we can ask, what will happen if we set the numbers using more sophisticated heuristics such as *displacement-adjusted virtual nesting* (DAVN) or *prorated expected marginal seat revenue* (PEMSR)? (See Talluri and van Ryzin 2004, pp. 102–106.) Although the DLP is simpler in that it does not require distributional information apart from the expected demand, we think that the ubiquity of DAVN and PEMSR make their study interesting. Second, in realistic settings, demand is estimated and control is applied simultaneously. Extending the work here to a case of unknown demand rate would be a significant step in this direction. In particular, we are interested to know whether DLP-based heuristics are equally robust with respect to estimation error. Finally, it would be useful to have heuristics that are more decentralized and yet have a good performance to mitigate the complexity of having to solve a very large LP very frequently.

Supplemental Material

Supplemental material to this paper is available at http://dx.doi.org/10.1287/opre.2013.1216.

Acknowledgments

The authors thank the three anonymous reviewers and the area editor whose valuable insights and suggestions improve both the organization and the clarity of the materials presented in the paper.

References

Berstimas D, Popescu I (2003) Revenue management in a dynamic network environment. *Transportation Sci.* 37(3):257–277.

Brumelle SL, McGill JI (1993) Airline seat allocation with multiple nested fare classes. *Oper. Res.* 41(1):127–137.

Chen L, Homem-de-Mello T (2010) Re-solving stochastic programming models for airline revenue management. *Ann. Oper. Res.* 177(1): 91–114.

Cooper WL (2002) Asymptotic behavior of an allocation policy for revenue management. *Oper. Res.* 50(4):720–727.

Jasin S, Kumar S (2012) A re-solving heuristic with bounded revenue loss for network revenue management with customer choice. *Math. Oper. Res.* 37(2):313–345.

Maglaras C, Meissner J (2006) Dynamic pricing strategies for multi-product revenue management problems. *Manufacturing Service Oper. Management* 8(2):136–148.

Perakis G, Roels G (2010) Robust controls for network revenue management. *Manufacturing Service Oper. Management* 12(1):56–76.

Reiman MI, Wang Q (2008) An asymptotically optimal policy for a quantity-based network revenue management problem. *Math. Oper. Res.* 33(2):257–282.

Secomandi N (2008) An analysis of the control-algorithm re-solving issue in inventory and revenue management. *Manufacturing Service Oper. Management* 10(3):468–483.

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

Talluri K, van Ryzin G (2004) *The Theory and Practice of Revenue Management* (Springer, New York).

Stefanus Jasin is an assistant professor in the Stephen M. Ross School of Business at the University of Michigan, Ann Arbor. His research interests include stochastic optimization, online learning and optimization, revenue management, and dynamic pricing.

Sunil Kumar is dean and professor of operations management at the Booth School of Business, University of Chicago. Before joining Booth, he was on the faculty of the Stanford University Graduate School of Business, where he was the Fred H. Merrill Professor of Operations, Information and Technology, and he also served as senior associate dean for academic affairs. His research interests include performance evaluation and control of manufacturing systems, service operations, and communication networks. He also studies the application of optimization methods and control theory to managerial problems.