Simulation-Based Booking Limits for Airline Revenue Management

Article *in* Operations Research · February 2005

DOI: 10.1287/opre.1040.0164 · Source: DBLP

Dimitris Bertsimas

Massachusetts Institute of Technology

OPERATIONS RESEARCH
Vol. 53, No. 1, January–February 2005, pp. 90–106
ISSN 0030-364X | EISSN 1526-5463 | 05 | 5301 | 0090

DOI 10.1287/opre.1040.0164

Simulation-Based Booking Limits for Airline Revenue Management

Dimitris Bertsimas, Sanne de Boer
Operations Research Center, Massachusetts Institute of Technology, 77 Massachusetts Avenue, E40-130, Cambridge, Massachusetts 02139
{dbertsim@mit.edu, sanne@alum.mit.edu}

Deterministic mathematical programming models that capture network effects play a predominant role in the theory and practice of airline revenue management. These models do not address important issues like demand uncertainty, nesting, and the dynamic nature of the booking process. Alternatively, the network problem can be broken down into leg-based problems for which there are satisfactory solution methods, but this approach cannot be expected to capture all relevant network aspects. In this paper, we propose a new algorithm that addresses these issues. Starting with any nested booking-limit policy, we combine a stochastic gradient algorithm and approximate dynamic programming ideas to improve the initial booking limits. Preliminary simulation experiments suggest that the proposed algorithm can lead to practically significant revenue enhancements.

Subject classifications: simulation: applications; inventory: perishable items; transportation: airlines.
Area of review: Manufacturing, Service, and Supply Chain Operations.
History: Received January 2001; revisions received November 2001, December 2003; accepted December 2003.

1. Introduction

After the deregulation of the airline industry in the 1970s, airlines started offering a variety of fares for seats in the same cabin. The question of how many seats should be offered at each different rate is commonly referred to as the airline *revenue management* (RM) problem. The economic importance of RM is illustrated by Delta Airlines' estimate that selling only one seat per flight at full rather than at discount rate adds over $50 million to its annual revenues (Cross 1997).

The problem of optimizing the passenger mix on a single-leg flight has received a lot of attention in the academic literature. The early model by Littlewood (1972) for the basic case with only two fare classes is based on the concept of the *expected marginal seat revenue* (EMSR). This is the expected marginal revenue of holding an additional seat for a certain fare class. Belobaba (1987, 1989) generalized this approach to a heuristic booking policy for multiple fare classes, which was extended by Belobaba and Weatherford (1996) to incorporate sell-ups. Brumelle et al. (1990) examine the impact of demand correlation. More recent models use stochastic dynamic programming (SDP) to determine an optimal policy, e.g., Wollmer (1992), Brumelle and McGill (1993), Robinson (1995), and Lee and Hersh (1993). Lautenbacher and Stidham (1999) provide a unified model of this and related work. These models have been extended to incorporate important practical issues such as overbooking, cancellations, and no-shows (e.g., Chatwin 1996, 1998; Subramanian et al. 1999). See Weatherford (1998) for a discussion of the issues related to RM and McGill and van Ryzin (1999) for a comprehensive overview of the literature in this field.

While booking policies such as Belobaba's EMSR have proven to be very profitable, it was recognized in the early 1990s that an airline should aim to optimize its bookings over its network as a whole, rather than on each flight leg in isolation. Williamson (1992) used simulation to show that explicitly addressing the network aspect of the revenue management problem leads to a significant increase of expected revenue over leg-based methods. The implementation of such origin-destination based policies is still in progress at many major airlines (e.g., Saranathan et al. 1999 of United Airlines and Pagé 1999 of Air Canada).

Although in theory the network aspect can easily be added to an SDP model (e.g., Gallego and van Ryzin 1997), in practice this is infeasible. Because the size of the state space is determined by the number of available seats on each flight of the network, the number of classes, and the number of possible passenger itineraries, the problem size explodes even for a moderate-size airline. This "curse of dimensionality" of the SDP method necessitates the development of alternative methods.

Mathematical programming (MP) models are especially well suited to incorporate network effects, i.e., to recognize which itineraries contribute most to the airline's revenue. Dimensionality problems, however, necessitate the use of linear optimization models, which we discuss in the next section. The solution of these models can be used to implement two different forms of booking control that are used in practice, one based on *booking limits* and the other based on *bid prices*. The focus of this research is on booking-limit control.

The MP models that have been proposed, while effective in addressing network effects, are deterministic, static, and partitioned, thus ignoring the stochastic and dynamic nature of the demand and the nested character of booking-limit control in a network. As a result, policies based on the MP seat allocations may only capture part of the potential revenue gain. In practice, most RM systems only use the dual solution of the MP to heuristically account for displacement costs. The network problem can then be broken down into leg-based problems for which there are satisfactory solutions. However, this approach cannot be expected to capture all relevant network aspects.

Our overall objective in this research is to propose a new way to calculate booking limits that takes into account the stochastic and dynamic nature of the demand and the nested character of booking-limit control in a network, is practically feasible for realistically sized problems, and leads to revenue enhancements. The basic idea of our approach is to approximate the expected revenue of any nested booking-limit policy by simulating the booking process until the current booking limits are up for revision. From there, a value function estimate is used to approximate the revenue that would have been generated in the remaining part of the booking period. This revenue approximation scheme is embedded in a stochastic gradient-type algorithm to iteratively improve any initial set of booking limits. In particular, these could have been determined by any of the methods referred to above. The contributions of this research are:

1. We propose a new approach to airline RM that, building upon RM research in the last decade, combines linear optimization models that successfully address the network effects with the newer ideas in this field of simulation-based optimization and approximate dynamic programming. To the best of our knowledge, our method is the first to *simultaneously* address nesting as well as the stochastic and dynamic nature of demand in a network environment.
2. We provide computational evidence that indicates that the proposed method can lead to revenue enhancements over current methodology and is computationally feasible for realistically sized problems.

This paper is structured as follows. In §2, we discuss the state of the art of origin-destination based seat inventory control. We present the MP models that have been proposed for this purpose and discuss booking-limit and bid-price control. In particular, this section contains the first step of our approach to the RM problem by determining an initial booking-limit policy. In §3, we present the second part of our overall approach, which uses simulation-based optimization. In §4, we provide computational evidence of the strength of our approach and its practical applicability. Section 5 summarizes our conclusions.

2. Mathematical Programming-Based Approaches to Network Revenue Management

In what follows, we use the term *booking class* to refer to the typical combination of origin, destination, and fare class, which we index by *odf*. Glover et al. (1982) proposed an integer programming model to determine the number of seats that should be available to each booking class, with the following linear programming relaxation:

$$\begin{aligned} \text{maximize} \quad & \sum_{odf} f_{odf} x_{odf} \\ \text{subject to} \quad & x_{odf} \leqslant E[D_{odf}] \quad \forall\, odf, \\ & \sum_{odf \in S_l} x_{odf} \leqslant C_l, \quad l = 1, \ldots, L, \\ & x_{odf} \geqslant 0. \end{aligned} \tag{1}$$

Here and in the remainder of this paper, f_{odf}, $E[D_{odf}]$, and x_{odf} are the fare, expected demand, and the number of seats allocated to booking class *odf*, respectively; C_l is the seat capacity of leg l, L is the total number of legs, and S_l is the set of booking classes that travel through leg l. Note that Model (1) only encompasses a single flight complex in a single time window. Many airlines operate several flights per day between city pairs and offer multiple connections between the same origin and destination. The indexing of flights and *odf*s can easily be extended to incorporate this aspect.

It is well known that if the itinerary between any origin and destination is not fixed and the airline can route its passengers over its network, Model (1) can be reformulated as a network flow problem. In practice, however, this is not the case, and even if the demand forecasts are integer, this model does not necessarily have an integer solution. Given that it is computationally unattractive to solve an integer program, we focus on solving the relaxation. Its solution can be used to implement two different types of booking policies: one based on booking limits and another based on bid prices.

2.1. Booking-Limit Control

Let x^*_{odf} be an optimal solution of Model (1). The policy represented by the model allocates up to $\lfloor x^*_{odf} \rfloor$ seats to class *odf*. A disadvantage of such a policy is that it partitions the seat capacity of the airline's network. When seats for a booking class are sold out, additional booking requests for this class will be declined, which can lead to lost revenue. Intuitively, seats allocated to the least profitable booking classes should be made available to more profitable classes as well, which is called *nesting*. Smith and Penn (1988) of American Airlines proposed the *Displacement Adjusted Virtual Nesting* scheme to implement nested booking-limit control in a network, which is outlined below. The operational policy was not specified, but we include both methods that are used in practice (steps 4′ and 4″). (We thank Dr. Peter Belobaba for insightful comments on these methods.)

Displacement Adjusted Virtual Nesting (DAVN)

Step 1 (Nesting order). Let $\{\pi_l \mid l = 1, \ldots, L\}$ be an optimal set of dual prices corresponding to the capacity constraints of Model (1). Calculate the displacement adjusted leg revenues

$$\bar{f}^l_{odf} = f_{odf} - \sum_{\{l' \neq l:\, odf \in S_{l'}\}} \pi_{l'} \quad (l = 1, \ldots, L).$$

This is a measure of profitability that determines a nesting order of booking classes on each leg.

Step 2 (Clustering). Cluster booking classes of similar adjusted leg revenue into a manageable number (in practice about 5–10) of *leg buckets* on each leg. These buckets are ordered from high to low, such that bucket 1 on leg l contains the most profitable booking classes on this leg, followed by bucket 2, and so forth. In what follows, let B^l_i be the set of booking classes *odf* in bucket i on leg l, let I^l_{odf} be the bucket of booking class *odf* on leg l, let N_l be the number of buckets defined on leg l, and let $N = \sum_{l=1}^{L} N_l$ be the total number of leg buckets in the network. For completeness, the clustering algorithm we used in this research is presented in the appendix.

Step 3 (Booking-limit calculation). Define a booking limit b^l_i for each leg bucket i on leg l, such that $b^l_i \geqslant b^l_{i+1}$ for $i = 1, \ldots, N_l - 1$ and $b^l_1 = C_l$, because in the absence of overbooking, the sale of booking classes in the highest bucket should only be restricted by capacity. In this research, we have considered two booking-limit calculation methods, one based on the LP seat allocations (LP-BL) originally proposed by Williamson (1992), and the other (EMSR-BL) based on the well-known EMSRb method proposed by Belobaba (1992), which uses the displacement adjusted leg revenues. For completeness, both methods are described in the appendix.

Step 4′ (Operational policy 1: Standard nesting). Interpret b^l_i as the maximum number of seats we are willing to sell on leg l for booking classes mapped into bucket i *or lower*. Let $d^l_{i:N_l}$ denote the accepted demand for buckets i through N_l (i.e., bucket i and lower) on leg l since the most recent booking-limit calculation. These variables are updated continuously between reoptimizations of the booking policy. Consider a request for booking class *odf*. If on any of its legs l: $odf \in S_l$ the booking limit for bucket $i = I^l_{odf}$ or any higher bucket $i < I^l_{odf}$ has been reached (i.e., $d^l_{i:N_l}$ equals b^l_i), the booking request is declined; otherwise, it is accepted and the accepted demand variables are updated accordingly ($d^l_{i:N_l} \leftarrow d^l_{i:N_l} + 1$ for all legs l: $odf \in S_l$ and inventory buckets $i \leqslant I^l_{odf}$).

Step 4″ (Operational policy 2: Theft nesting). Stops selling seats on leg l to booking classes mapped into bucket i after b^l_i seats have been sold on this leg, regardless of to whom. Let d^l denote the accepted demand for seats on leg l (i.e., for $odf \in S_l$) since the most recent booking-limit calculation. These variables are updated continuously between reoptimizations of the booking policy. Consider a request for booking class *odf*. If on any of its legs l: $odf \in S_l$ the booking limit for bucket $i = I^l_{odf}$ has been reached (i.e., d^l equals b^l_i), the booking request is declined; otherwise, it is accepted and the accepted demand variables are updated accordingly ($d^l \leftarrow d^l + 1$ for all legs l: $odf \in S_l$).

Note that theft nesting is more restrictive than standard nesting because low buckets will close earlier in the booking process. This will lead to a higher yield, but a lower load factor. To our knowledge, industry experts differ in opinion about which policy is better, and both may be found in practice.

As far as we know, DAVN in combination with EMSR-BL is currently being used by only a handful of airlines worldwide. The term *virtual nesting* was chosen by Smith and Penn (1988) to reflect that the availability of a booking class is never stored in the system, which given the dimensions of the problem is impractical, but can be determined when needed from the leg-bucket availabilities. As they point out, it is not an optimization technique, but a control framework that allows a reservation system to approximate market class (*odf*) control. In §3, we propose a new method to calculate booking limits within the framework of DAVN.

2.2. Bid-Price Policy

An alternative booking policy is based on the general idea of rejecting a request unless its fare exceeds the expected opportunity cost of not being able to sell the requested seats at a later time, which can be seen as the *bid price* of a particular itinerary. There are several ways to approximate the "true" bid prices of itineraries on a flight network. Simpson (1989) proposed the following method based on the solution of Model (1).

Bid-Price Policy (BP)

Step 1. Solve Model (1) and, using an optimal set of dual variables π_l for each leg l, calculate the net contributions to network revenue $\bar{f}_{odf} = f_{odf} - \sum_{l:\, odf \in S_l} \pi_l$.

Step 2. If an incoming booking request has $\bar{f}_{odf} \geqslant 0$, accept the request; otherwise, reject it.

One obvious disadvantage of this policy is that once a booking class is open to bookings, there is no limit on the number of booking requests that are accepted for this class. This induces the risk of flights filling up with passengers from booking classes that only marginally contribute to network revenue. To prevent this, it is essential that the bid prices are updated frequently during the booking period. An alternative solution, which we have not considered in this research, is to use dynamic bid prices that adjust as capacity is consumed. Van Ryzin (1998) argues that dynamic bid prices offer essentially the same level of control over the booking process as booking limits do. Additional issues related to bid-price control are discussed in Williamson (1992), Talluri and van Ryzin (1998), Chen et al. (1999), and Bertsimas and Popescu (2000).

2.3. Modeling Issues

The LP-based booking limits clearly need not be optimal for a given virtual nesting scheme, because they are based on an inaccurate representation of the RM problem. Model (1) is deterministic, static, and determines a seat partitioning, while in reality airline demand is stochastic, the booking process is dynamic, and seat inventory control is nested. There have been some proposals to address the first issue and include stochastic demand. Wollmer (1986) proposes an optimization model involving binary decision variables $x_{odf}(i)$ that indicate whether i or more seats in the network are allocated to a particular booking class. The objective coefficient of $x_{odf}(i)$ is the expected marginal revenue of allocating an additional ith seat to the booking class *odf*. The drawback of this model is the large number of decision variables, which severely limits its practical applicability. De Boer et al. (2002) propose a scenario-based stochastic programming model to overcome these dimensionality problems, while Talluri and van Ryzin (1999) propose a randomized linear programming approach. However, numerical results by Williamson (1992) and de Boer et al. (2002) show that, in fact, the deterministic model gives the best input for the LP-BL heuristic. This suggests that demand uncertainty and nesting need to be addressed simultaneously in the booking-limit calculation step.

Curry (1990) proposes a two-step algorithm in this direction. First, a linear programming model with a piecewise linear approximation of the expected revenue as a function of the seat partitioning is used to determine seat allocations for each *itinerary* in the network. Then, given these allocations, nested booking limits are calculated for each *fare class* based on a single-leg method that incorporates demand uncertainty. However, Curry's method is not suited to nest all *booking classes* on a flight leg simultaneously, which, given the oftentimes large fare differences between itineraries with common flight legs, is desirable.

The EMSR-BL method is another way to approach this problem. The booking limits are calculated for each leg individually, but the network aspect is to some extent captured by the displacement adjusted leg revenues. However, by breaking up the problem we do lose some low-level characteristics that could affect the optimal set of booking limits for a given virtual nesting scheme. For instance, the optimal mix of local and connecting traffic should depend on the probability of an accepted connecting passenger displacing local passengers on all his flight legs simultaneously, which cannot be captured properly by an average displacement measure. In addition, the EMSR method is based on the assumption that there is a strict low-before-high arrival order of booking requests for different *fare classes*. Although this is generally believed to be a workable assumption for practical RM purposes, there is no reason that there would be a particular arrival order of booking requests for the different *itineraries* that make up the buckets in the virtual nesting scheme. Note that in that case, the choice between standard and theft nesting affects revenue, but this is not included in the model. This may affect the performance of the EMSR-based booking limits in a network environment as well.

2.4. Other Methods

Several other methods have been proposed to implement booking control in a network. The first such approach, somewhat similar to DAVN, is to break up the network problem into leg-based problems by allocating the revenue of a multileg flight over its legs, which is called *prorating*. For these subproblems, we do have satisfactory stochastic and dynamic solution methods. Smith and Penn (1988) and Williamson (1992) consider prorating based on the mileage of the individual flight legs, while Bratu (1999) studies an iterative prorating algorithm based on the expected marginal seat revenue of the last seat on each leg. However, despite some promising test results by Bratu, there is no intuitive explanation of why the prorating approach should correctly take into account network effects.

The second approach is approximate dynamic programming (ADP). Given that exact SDP is not applicable due to dimensionality problems, Bertsimas and Popescu (2000) propose to approximate the SDP value function by Model (1). This is a form of *certainty equivalent control*, which is one of the heuristic solution techniques for complex dynamic programming problems discussed in Bertsekas (1995).

Chen et al. (1998) show that the objective value of Model (1) is in fact an upper bound on the optimal expected revenue, while any nonnested probabilistic model such as Wollmer's (1986) provides a lower bound. They argue that the opportunity costs of selling a particular itinerary are actually underestimated by the stochastic model, whereas they are overestimated by the deterministic model. This insight leads to an algorithm in which the revenue of every booking request is compared to both estimates of the opportunity costs. The idea is to diminish the number of wrong decisions resulting from the biased estimates of the opportunity costs that are based on any such single model. Simulation experiments indicate that this approach indeed leads to higher revenues than the BP policy, but further numerical testing is required to gather more conclusive evidence.

3. Simulation-Based Optimization

In this section, we show how to improve the booking limits of any particular BL policy (for instance, EMSR-BL or LP-BL) by taking into account the stochastic and dynamic nature of the demand. We use a combination of simulation-based optimization (a stochastic gradient algorithm) and approximate dynamic programming ideas.

The idea of simulation-based optimization in a revenue-management context was introduced by Robinson (1995). For the single-leg problem under the assumption that demand for different fare classes arrives sequentially, he derives optimality conditions for the booking limits. He then proposes to solve these using Monte Carlo integration, based on simulation of the demand. Van Ryzin and McGill (2000) propose an adaptive method to solve the same set of optimality conditions, but using historical booking data instead of simulated demand. The convergence of this method is shown using stochastic approximation theory. Karaesmen and van Ryzin (1998) develop a numerical algorithm to determine joint overbooking levels for partly substitutable inventory classes. This takes the use of simulation one step further, because no optimality conditions are known for this problem.

The reason that simulation-based optimization has not been applied more generally is most likely the enormous size of the problem. The combination with mathematical programming models that we propose here significantly reduces the problem size by efficiently dealing with the combinatorial aspects of the problem, i.e., the nesting order. As Williamson (1992, p. 127) nicely puts it: "Without first knowing the "correct" nesting hierarchy of different *odf*'s over a network, a network optimization which explicitly accounts for nesting of *ODF*'s is, to date, both theoretically infeasible and impractical."

3.1. Problem Definition

Model (1) is a static optimization model in the sense that it determines a booking policy that is implemented until flight departure. However, as the booking process proceeds, the airline finds out the actual realization of demand. This information can be used to improve the booking policy, for instance, by holding additional seats for profitable booking classes with higher than average demand. Hence, in practice the booking policy is revised several times, most often in overnight runs of the optimization algorithm. It is important that these future revisions are taken into account when determining a booking policy. For instance, such revisions allow an airline to protect more seats for business travelers in the early stages of the booking period because it will be able to offer additional discount seats later on if necessary. We have added a numerical example in §4.3 that illustrates this point.

We assume that the booking period is divided into T time windows, not necessarily of the same length. Whenever the booking process enters a new time window t, the booking policy is reoptimized. During each time window, the nesting order and bucket mapping are fixed and based on Model (1). In what follows, let $\mathbf{b}^t$ be the set of booking limits used in time window t, let $\mathbf{C}^t$ be the remaining capacity vector at the beginning of time window t, and let $q^t(\mathbf{b}^t, \mathbf{C}^t)$ be the corresponding revenue generated during time window t. Let $Q^t(\mathbf{b}^t, \mathbf{C}^t)$ be the total revenue generated during time windows t to T if the airline uses the booking limits $\mathbf{b}^t$ in time window t and the optimal (maximizing expected revenue) booking-limit policy thereafter. Note that, given the demand model and the virtual nesting scheme, this notion is well defined, because the number of allowable sets of booking limits is finite (cf. §2.1). Finally, let $R^t(\mathbf{C}^t)$ be the total revenue generated from time window t to T if the airline implements the optimal booking-limit policy during each of these, given the intermediate state $\mathbf{C}^t$ of the booking process. By definition, $q^t(\cdot)$, $Q^t(\cdot)$, and $R^t(\cdot)$ are all random functions whose values depend on the realization of the demand process. At the beginning of time window t, the airline needs to solve

$$E[R^t(\mathbf{C}^t)] \equiv \max_{\mathbf{b}^t} E[Q^t(\mathbf{b}^t, \mathbf{C}^t)] \tag{2}$$

$$\equiv \max_{\mathbf{b}^t} E[q^t(\mathbf{b}^t, \mathbf{C}^t) + R^{t+1}(\mathbf{C}^{t+1})] \tag{3}$$

for the optimal set of booking limits, where the expectation is taken with respect to the demand process. Note that (3) explicitly formulates the booking-limit optimization problem as a dynamic program.

For any realistic model of demand, $E[Q^t(\mathbf{b}^t, \mathbf{C}^t)]$ cannot be expressed in closed form and needs to be evaluated numerically. We propose to simulate the booking process to determine the revenue $q^t(\cdot)$ generated during time window t, which directly depends on $\mathbf{b}^t$. We then use an estimate of the *value function* $E[R^{t+1}(\mathbf{C}^{t+1})]$ to account for the revenue that would have been generated in the remainder of the booking period. As we shall see, we can then attempt to solve problem (2) by a stochastic gradient algorithm. Alternatively, the booking process could be simulated over all remaining time windows to evaluate $Q^t(\mathbf{b}^t, \mathbf{C}^t)$ directly. However, then the booking limits would have to be recalculated multiple times during each simulation run, which for a large number of runs and realistically sized problems would be intractable.

In §3.2, we define a computationally efficient stochastic gradient algorithm that approximates a solution to problem (2). In §3.3, we develop a recursive algorithm to estimate the value functions $E[R^t(\mathbf{C}^t)]$ that are needed for this. In §3.4, we combine these algorithms and propose a simulation-based method for booking-limit calculation.

3.2. The Stochastic Gradient Algorithm

In this section, we propose a stochastic gradient algorithm to approximate a solution to problem (2), given an approximation of $E[R^{t+1}(\mathbf{C}^{t+1})]$. EMSR-BL or LP-BL provide an initial solution. The algorithm iteratively improves the set of booking limits, using numerical estimates of the first finite differences of $E[Q^t(\mathbf{b}^t, \mathbf{C}^t)]$. First, we briefly review the principles of stochastic gradient algorithms to prepare the ground for what follows. Then, we adapt the stochastic gradient algorithm to problem (2).

3.2.1. A Generic Stochastic Gradient Algorithm. Let Ω be the set of realizations of a random process. Let $F(\mathbf{x}, \boldsymbol{\omega})$ be a function of some variable $\mathbf{x} \in \mathbb{R}^m$, whose value depends on the random outcome $\boldsymbol{\omega} \in \Omega$. Possibly, the function $F(\cdot)$ can only be evaluated using simulation. In the RM context, $\mathbf{x}$ is a vector of parameters that characterize a policy (either booking limits or bid prices), $F(\cdot)$ is the revenue as a function of these parameters, while $\boldsymbol{\omega}$ is the particular realization of the stochastic and dynamic demand. We are interested in maximizing $E_{\boldsymbol{\omega}}[F(\mathbf{x}, \boldsymbol{\omega})]$ over $\mathbf{x} \in X$, where $X \subseteq \mathbb{R}^m$ denotes the set of feasible solutions. The stochastic gradient algorithm is as follows.

Generic Stochastic Gradient Algorithm

Step 0. Pick a starting value $\mathbf{x}^0 \in X$ and let $k = 0$.

Step 1. Let $\mathbf{x}^{k+1} = \pi_X[\mathbf{x}^k + \rho_k \boldsymbol{\xi}^k]$, where $\boldsymbol{\xi}^k \in R^m$ is defined by

$$\xi_i^k = \sum_{j=1}^{n} \frac{F(\mathbf{x}^k + \Delta_k \mathbf{e}^i, \boldsymbol{\omega}^{kj}) - F(\mathbf{x}^k, \boldsymbol{\omega}^{kj})}{\Delta_k} \quad (i = 1, \ldots, m),$$

and $\pi_X[\mathbf{x}]$ denotes the projection of a vector $\mathbf{x}$ on the feasible region X.

Step 2. Let $k = k + 1$. Return to Step 1.

Here, $\boldsymbol{\xi}^k$ is a finite difference approximation of the gradient $\nabla E_{\boldsymbol{\omega}}[F(\mathbf{x}^k, \boldsymbol{\omega})]$ based on n independent samples $\boldsymbol{\omega}^{kj}$ of the random process, Δ_k is small positive scalar, and ρ_k is the stepsize that typically is decreasing in k. Note that this algorithm is well defined, even if the objective function $E_{\boldsymbol{\omega}}[F(\mathbf{x}, \boldsymbol{\omega})]$ is not differentiable. Under certain regularity conditions for the function $F(\cdot)$, the sequences ρ_k and Δ_k, and the feasible region X, this algorithm can be shown to converge to an optimal solution with probability 1 (e.g., Ermoliev 1988, Gaivoronski 1988). However, as discussed in §3.2.3, these conditions are not met for our particular application.

3.2.2. A Stochastic Gradient Algorithm for Booking-Limit Improvement. We now adapt the stochastic gradient algorithm to problem (2). To facilitate notation, we have suppressed the dependence of $Q^t(\mathbf{b}^t, \mathbf{C}^t)$ on the time window t and the remaining capacity $\mathbf{C}^t$ in this subsection.

By our definition of booking-limit control, even if the integrality of booking limits is relaxed, $E[Q(\mathbf{b})]$ is a function of the integral part of $\mathbf{b}$ only. As a result, the objective function in problem (2) has discontinuities at the integer values of $\mathbf{b}$ and is not differentiable. In our implementation of the stochastic gradient algorithm, we therefore have to work with numerical estimates of the first finite differences

$$\Delta_i^l E[Q(\mathbf{b})] = E[Q(\mathbf{b}+\mathbf{e}_i^l)] - E[Q(\mathbf{b})] \quad \text{for } 1 \leqslant i \leqslant N_l, \quad (4)$$

where $\mathbf{e}_i^l$ is the unit vector corresponding to b_i^l. These can be interpreted as the expected revenue change when a booking limit is increased by exactly one seat. Note that $\Delta_1^l E[Q(\mathbf{b})] = 0$, because we assume that $b_1^l = C_l$. Even if this booking limit is increased, this does not affect which booking requests are accepted due to the capacity constraint. The vector of first finite differences $\Delta E[Q(\mathbf{b})]$ of the expected revenue function is defined by

$$\Delta E[Q(\mathbf{b})] = [\Delta_i^l E[Q(\mathbf{b})]]_{l=1,\ldots,L;\, i=1,\ldots,N_l}.$$

To estimate $\Delta E[Q(\mathbf{b})]$, we could simply randomly generate n sequences of booking requests, evaluate $Q(\mathbf{b})$ and $Q(\mathbf{b}+\mathbf{e}_i^l)$ for each of these for all buckets i, l, and substitute the averages into (4). Cf. Step 2 of the stochastic gradient algorithm. However, accurate estimation of the first finite differences requires a large number of simulation runs and the evaluation of nN booking-limit policies may be time consuming. We propose a more efficient approach here, based on the insight that the increase of a single booking limit does not necessarily affect which booking requests are accepted. Demand might simply be too low to be constrained by any booking limit, an inventory bucket might be closed for further bookings because the booking limit of a higher bucket is reached, or accepting a booking request might violate the booking limits of several inventory buckets at the same time, in which case increasing just one of them makes no difference. Thus, in many cases $Q(\mathbf{b})$ equals $Q(\mathbf{b}+\mathbf{e}_i^l)$. This insight motivated the following algorithm to estimate $\Delta E[Q(\mathbf{b})]$.

Finite Differences Estimation Algorithm

Step 1. FOR $j = 1$ TO n

(1a) Simulate the booking process in time window t; that is, generate a sequence of time-ordered booking requests according to some demand model, such as the one defined in §4. Starting with the earliest one, these requests are then worked through one at a time, either accepting or rejecting them as dictated by some prespecified policy—in this case the booking limits $\mathbf{b}$—and the capacity constraints.

(1b) Let $\overline{Q(\mathbf{b}, j)}$ be the revenue estimate of using the booking limits $\mathbf{b}$ in time window t and some periodically revised booking-limit policy thereafter, based on the revenue generated in simulation j and the approximation of $E[R^{t+1}(\mathbf{C}^{t+1})]$.

(1c) Whenever a booking request is declined because the booking limit of exactly one inventory bucket, say B_i^l, would be violated, start keeping track of what would have happened if this booking limit would have been one seat higher *given the same future sequence of booking requests*. Note that the booking request in that case would have been accepted. Do this at most once for each inventory bucket. Let the resulting revenue estimate be denoted by $\overline{Q(\mathbf{b}+\mathbf{e}_i^l, j)}$.

(1d) If $\overline{Q(\mathbf{b}+\mathbf{e}_i^l, j)}$ has been defined, let $\overline{\Delta_i^l Q(\mathbf{b}, j)} = \overline{Q(\mathbf{b}+\mathbf{e}_i^l, j)} - \overline{Q(\mathbf{b}, j)}$. Otherwise, let $\overline{\Delta_i^l Q(\mathbf{b}, j)} = 0$.

END FOR

Step 2. Let $\overline{\Delta_i^l E[Q(\mathbf{b})]} = (1/n) \sum_{j=1}^{n} \overline{\Delta_i^l Q(\mathbf{b}, j)}$ be the final estimate of $\Delta_i^l E[Q(\mathbf{b})]$ and let $\overline{\Delta E[Q(\mathbf{b})]}$ denote the corresponding estimate of $\Delta E[Q(\mathbf{b})]$.

Note that we only generate n sequences of booking requests, because Step (1c) is simply a matter of "parallel bookkeeping." "Parallel simulations" only branch off the main simulation; hence, their number remains limited. The point is that for each realization of demand, we only evaluate the booking-limit policy $\mathbf{b}+\mathbf{e}_i^l$ if the result would differ from $\mathbf{b}$. This can be significantly more efficient than evaluating each of these N policies consecutively, for instance, given low demand. The sample size n needs to be tuned beforehand. Larger values give more accurate estimates at the cost of more computation time.

Using these estimates of the first finite differences, the booking limits will now be iteratively adjusted until no further improvement seems possible or the maximum number of iterations $k_{\max}$ is reached. Any feasible set of booking limits $\mathbf{b}$ has to satisfy the constraint

$$0 \leqslant b_{i+1}^l \leqslant b_i^l \leqslant b_1^l = C_l \quad \text{for } 2 \leqslant i \leqslant N_l - 1.$$

Experimental evidence has shown that booking limits should only be moderately increased during a single iteration of the stochastic gradient algorithm. For this reason, we have introduced an upper bound $\Delta_{\max}$ on the change of any particular booking limit in each iteration. To address these issues, we have modified the stochastic gradient algorithm as follows.

Numerical Booking-Limit Improvement Step

Step 0. Let $k=1$; let $\mathbf{b}_1$ be an initial set of booking limits. Let $\mathbf{b}_0=\mathbf{b}_1$ and $\overline{\Delta E[Q(\mathbf{b}_0)]}=0$.

Step 1. Estimate $\overline{\Delta E[Q(\mathbf{b}_k)]}$.

Step 2. FOR $l=1$ TO L
    FOR $i=2$ TO N_l
        IF $\overline{\Delta_i^l E[Q(\mathbf{b}_k)]}>0$ AND $\overline{\Delta_i^l E[Q(\mathbf{b}_{k-1})]}\geqslant 0$
         (clearly increase the booking limit)
            $\Delta_i^l=\min\{\text{int}(\overline{\Delta_i^l E[Q(\mathbf{b}_k)]}\rho_k+0.5),\Delta_{\max}\}$
        ELSE IF $\overline{\Delta_i^l E[Q(\mathbf{b}_k)]}<0$ AND
         $\overline{\Delta_i^l E[Q(\mathbf{b}_{k-1})]}\leqslant 0$ (decrease the booking limit)
            $\Delta_i^l=\max\{\text{int}(\overline{\Delta_i^l E[Q(\mathbf{b}_k)]}\rho_k-0.5),-\Delta_{\max}\}$
        ELSE (next booking limit between current and previous solution)
            $\alpha_i^l=|\overline{\Delta_i^l E[Q(\mathbf{b}_k)]}|/(|\overline{\Delta_i^l E[Q(\mathbf{b}_k)]}| + |\overline{\Delta_i^l E[Q(\mathbf{b}_{k-1})]}|)$ (with $\frac{0}{0}\triangleq 0$)
            $[\Delta_i^l]_{prev}=(b_i^l)_k-(b_i^l)_{k-1}$
            $\Delta_i^l=\text{int}(-\alpha_i^l*[\Delta_i^l]_{prev} - 0.5*\text{sign}([\Delta_i^l]_{prev}))$
        ENDIF
        $(b_i^l)_{k+1}=(b_i^l)_k+\Delta_i^l$
    END FOR
END FOR

Step 3. Make new solution feasible:
FOR $l=1$ TO L
    FOR $i=2$ TO N_l
        $(b_i^l)_{k+1}=\min\{(b_i^l)_{k+1},(b_{i-1}^l)_{k+1}\}$
        $(b_i^l)_{k+1}=\max\{(b_i^l)_{k+1},0\}$
    END FOR
END FOR

Step 4. IF $\Delta_i^l=0$ for all leg-buckets OR $k=k_{\max}$
    TERMINATE
ELSE
    $k=k+1$
    GOTO Step 1
ENDIF

The starting point $\mathbf{b}_1$ can be determined by EMSR-BL or LP-BL, whichever works best. The stepsize functions ρ_k, $\Delta_{\max}$, and $k_{\max}$ are optimization parameters that need to be tuned, in conjunction with the number of samples n for the first differences estimation algorithm. For instance, our experience with this algorithm suggests that because larger values of n give more accurate estimations of the first finite differences, this allows a larger stepsize ρ_k. Given that there are no theoretical performance guarantees for this algorithm, extensive tuning is especially important.

3.2.3. Theoretical Considerations. Given the discrete nature of booking limits, the expected revenue function $E[Q^t(\mathbf{b}^t,\mathbf{C}^t)]$ is nonsmooth and nondifferentiable. Attempting to solve this type of optimization problem with a stochastic gradient method has no theoretical justification, but is a heuristic based on the intuition behind gradient descent methods. The algorithm based on first finite differences is well defined, but does not necessarily converge (Ermoliev 1988). For this reason, we have introduced a maximum on the number of iterations.

Even if the algorithm would converge, this would not necessarily be at a "local optimum." To see this, consider a two-leg flight that only carries through-passengers at two different rates. Assume that the booking limit for the discount rate is the same on both legs. Then, given the BL policy of §2.1, increasing only one of these booking limits will not affect ticket sales. Thus, both first finite differences are zero and the random search will terminate, while increasing both booking limits simultaneously might have increased expected revenue. However, this is an unrealistic example given the absence of local traffic, and we do not expect problematic cases like this to occur in practice.

Because the expected revenue function may not be convex, the final set of booking limits may strongly depend on the starting point. We have found that the EMSRb heuristic generally gave a better starting point than the LP solution, especially for large-scale examples (see §4.4).

Summarizing, the algorithm cannot be guaranteed to terminate at an optimal set of booking limits for a given virtual nesting scheme, but intuition and practical experience gained in the experiments of the next section suggest that they can at least be improved significantly.

3.3. Estimation of the Value Function

In this section, we propose an algorithm to estimate the value functions $E[R^t(\mathbf{C}^t)]$, which are used as input for the Numerical Booking-Limit Improvement Step. Because the dimension of the capacity vector can be large, we propose to evaluate the value function only on a small number of carefully selected discretization points. This set of data is then used to estimate the value function as a whole. We first define an algorithm to determine the discretization points and then propose an interpolation method.

3.3.1. Selection of the Discretization Points. The domain of the value function is the Cartesian product of

$$\{0,1,2,\ldots,C_1\}\otimes\{0,1,2,\ldots,C_2\}\otimes\cdots\otimes\{0,1,2,\ldots,C_L\},$$

where L, as before, denotes the number of legs. Hence, each dimension of the state-space corresponds to a specific leg in the network. A natural way of selecting the discretization points is picking a small number, say q, of capacity levels on each leg, and forming a complete L-dimensional grid. However, the number of such grid points would be exponential in L, which would be intractable for large networks. In addition, it is questionable whether the discretization points should really be homogeneously distributed over the state-space. For instance, at the end of the first time window, remaining capacity is likely to be relatively close to the original flight capacities, whereas at the beginning of the final time window, flights are more likely to be sold out. Hence, if we could somehow find the most likely location of the remaining capacity vector in the state-space at any given time, we could concentrate the discretization points in this area. This would enable us to get more accurate estimates of the value function at places where it matters, with relatively few discretization points. We propose the following algorithm for this purpose.

Discretization Points Selection Algorithm

Step 1. Simulate the booking process. Whenever a simulation run enters a new time window, use a heuristic such as EMSR-BL to determine a reasonable booking policy, and save the remaining capacity of each leg of the network.

Step 2. Using the empirical distribution of $\mathbf{C}^t$ obtained in Step 1, calculate the mean $\bar{C}_l^t$ and standard deviation $\sigma(C_l^t)$ of the remaining capacity of leg l at the beginning of time window t.

Step 3. For the approximation of the value function at the beginning of time window t, use the discretization points

$$\big\{\bar{\mathbf{C}}_{l,i}^t=(\bar{C}_1^t,\ldots,\bar{C}_{l-1}^t,\bar{C}_{l,i}^t,\bar{C}_{l+1}^t,\ldots,\bar{C}_L^t)\mid i=1,\ldots,q;\; l=1,\ldots,L\big\},$$

where

$$\bar{C}_{l,i}^t=\bar{C}_{l,\min}^t+\left\lfloor\frac{i}{q+1}(\bar{C}_{l,\max}^t-\bar{C}_{l,\min}^t)\right\rfloor \quad (i=1,\ldots,q;\, l=1,\ldots,L)$$

for

$$\bar{C}_{l,\min}^t=\max\{\bar{C}_l^t-\delta\cdot\sigma(C_l^t),0\}\quad(i=1,\ldots,q;\, l=1,\ldots,L),$$
$$\bar{C}_{l,\max}^t=\min\{\bar{C}_l^t+\delta\cdot\sigma(C_l^t),C_l^1\}\quad(i=1,\ldots,q;\, l=1,\ldots,L).$$

Here δ is a predetermined constant defining a confidence interval centered around $\bar{C}_l^t$, in which the discretization points $\bar{\mathbf{C}}_{l,i}^t$ in dimension l are chosen. Multiples of discretization points may occur for small σ but are removed from the set.

For each time window t, the algorithm creates qL discretization points on the axes of an artificial coordinate system in the state-space with origin $\bar{\mathbf{C}}^t=(\bar{C}_1^t,\ldots,\bar{C}_L^t)$, that covers the area where the remaining capacity vector is most likely to be. Working within this artificial coordinate system facilitates the definition of the piecewise linear and separable approximation of the value function that we have used in this research. The number of discretization points q and the size δ of the confidence interval can only be chosen by trial and error, given the usual trade-off between accuracy and computation time. In the numerical examples of §4 we have used $\delta=3$, as this (roughly) corresponds to a 99% confidence interval.

3.3.2. Interpolation of the Value Function. In this section, we propose a method to estimate the value function $E[R^t(\mathbf{C}^t)]$ by approximating its value on the set of discretization points. For simplicity, we have used a piecewise linear and separable approximation in this research. This is somewhat similar to the bid-price approach, which can be seen as a linear approximation of the value function. Our approach utilizes first- and second-order information, which should lead to more accurate estimates. First, we develop an efficient algorithm to estimate the first finite differences of the value function, which is again not differentiable. Then, we propose a linear interpolation method that is motivated by the concavity of the expected revenue as a function of remaining capacity on a single leg (given Poisson demand in the absence of group bookings, e.g., Lee and Hersh 1993). The proposed approach is recursive, in the sense that the estimation of $E[R^t(\mathbf{C}^t)]$ requires that an estimate of $E[R^{t+1}(\mathbf{C}^{t+1})]$ is already available.

We propose the following algorithm to estimate the value function $E[R^t(\mathbf{C}^t)]$ and its first finite differences $\Delta_l E[R^t(\mathbf{C}^t)]$ for any given capacity vector $\mathbf{C}^t$.

Value Function and First Finite Differences Estimation Algorithm

Step 0. Use EMSR-BL or LP-BL (whichever works best) in combination with the Numerical Booking-Limit Improvement Step to determine a good set of booking limits $\mathbf{b}^t$.

Step 1. FOR $j=1$ TO n

(1a) Simulate the booking process in time window t. Let $\overline{R^t(\mathbf{C}^t,j)}$ be the estimate of the future revenue given capacity $\mathbf{C}^t$ at the start of time window t, based on the revenue generated in the simulation and the approximation of $E[R^{t+1}(\mathbf{C}^{t+1})]$.

(1b) For each leg l, determine what the revenue would have been if the capacity on this leg would have been decreased by one seat, *without changing the booking limits*. Let the revenue generated in that case be denoted by $\overline{R^t(\mathbf{C}_{l-}^t,j)}$.

(1c) Let the jth estimate of the lth first finite difference be $\overline{\Delta_l R^t(\mathbf{C}^t, j)} = \overline{R^t(\mathbf{C}^t, j)} - \overline{R^t(\mathbf{C}^t_{l-}, j)}$.

END FOR

Step 2. Let $\overline{E[R^t(\mathbf{C}^t)]} = (1/n)\sum_{j=1}^{n}\overline{R^t(\mathbf{C}^t, j)}$ and $\overline{\Delta_l E[R^t(\mathbf{C}^t)]} = (1/n)\sum_{j=1}^{n}\overline{\Delta_l R^t(\mathbf{C}^t, j)}$ be the final estimates of $E[R^t(\mathbf{C}^t)]$ and its lth first finite difference, respectively.

Note that the reduction of capacity on a leg might also affect the optimal set of booking limits, given the virtual nesting scheme. However, because reoptimization would be too time consuming, we have decided not to take this effect into account. The algorithm requires the generation of only n sequences of demand, because Step (1b) is again just a matter of parallel bookkeeping. The number of samples n is determined by tuning, the trade-off again being accuracy and computation time.

We now define the interpolation method. To simplify notation, we have suppressed the dependence on the time window t. Let $\overline{\mathbf{C}}_{l,1} < \overline{\mathbf{C}}_{l,2} < \cdots < \overline{\mathbf{C}}_{l,q}$ be the discretization points in dimension l, and let $\overline{C}_{l,i}$ be the lth coordinate of $\overline{\mathbf{C}}_{l,i}$. Use the estimation method outlined above to approximate the value function $E[R(\overline{\mathbf{C}}_{l,i})]$ and its lth first finite difference $\Delta_l E[R(\overline{\mathbf{C}}_{l,i})]$ for each discretization point $\overline{\mathbf{C}}_{l,i}$. Approximate $E[R(\overline{\mathbf{C}})]$ as well. Let $\mathbf{C} = (C_1, \ldots, C_L)$ and let $\mathbf{C}_l = \overline{\mathbf{C}} + (C_l - \overline{C}_l)\mathbf{e}_l$ be the projection of $\mathbf{C}$ on dimension l of the artificial coordinate system that contains the discretization points. Then, the value function at point $\mathbf{C}$ can be approximated by

Value Function Interpolation Algorithm

Step 1. For all legs $l = 1, \ldots, L$:

(1a) If $C_l \leqslant \overline{C}_{l,1}$, let $\overline{E[R(\mathbf{C}_l)]} \triangleq \overline{E[R(\overline{\mathbf{C}}_{l,1})]} - \overline{\Delta_l E[R(\overline{\mathbf{C}}_{l,1})]}(\overline{C}_{l,1} - C_l)$.

(1b) If $C_l \geqslant \overline{C}_{l,q}$, let $\overline{E[R(\mathbf{C}_l])} \triangleq \overline{E[R(\overline{\mathbf{C}}_{l,q})]} + \overline{\Delta_l E[R(\overline{\mathbf{C}}_{l,q})]}(C_l - \overline{C}_{l,q})$.

(1c) If $\overline{C}_{l,i} \leqslant C_l < \overline{C}_{l,i+1}$ $(1 \leqslant i < q)$, let

$$* \ \overline{E[R(\mathbf{C}_l)]}_{ub} \triangleq \min\left\{\begin{array}{l} \overline{E[R(\overline{\mathbf{C}}_{l,i})]} \\ \quad + \overline{\Delta_l E[R(\overline{\mathbf{C}}_{l,i})]}(C_l - \overline{C}_{l,i}), \\ \overline{E[R(\overline{\mathbf{C}}_{l,i+1})]} \\ \quad - \overline{\Delta_l E[R(\overline{\mathbf{C}}_{l,i+1})]}(\overline{C}_{l,i+1} - C_l) \end{array}\right\}$$

$$* \ \overline{E[R(\mathbf{C}_l)]}_{lb} \triangleq \overline{E[R(\overline{\mathbf{C}}_{l,i})]} + \frac{(C_l - \overline{C}_{l,i})}{\overline{C}_{l,i+1} - \overline{C}_{l,i}}\Big(\overline{E[R(\overline{\mathbf{C}}_{l,i+1})]} - \overline{E[R(\overline{\mathbf{C}}_{l,i})]}\Big)$$

$$* \ \overline{E[R(\mathbf{C}_l)]} \triangleq w\overline{E[R(\mathbf{C}_l)]}_{ub} + (1-w)\overline{E[R(\mathbf{C}_l)]}_{lb} \quad \text{for } 0 \leqslant w \leqslant 1.$$

Step 2. Let $\overline{E[R(\mathbf{C})]} = \overline{E[R(\overline{\mathbf{C}})]} + \sum_{l=1}^{L}(\overline{E[R(\mathbf{C}_l)]} - \overline{E[R(\overline{\mathbf{C}})]})$.

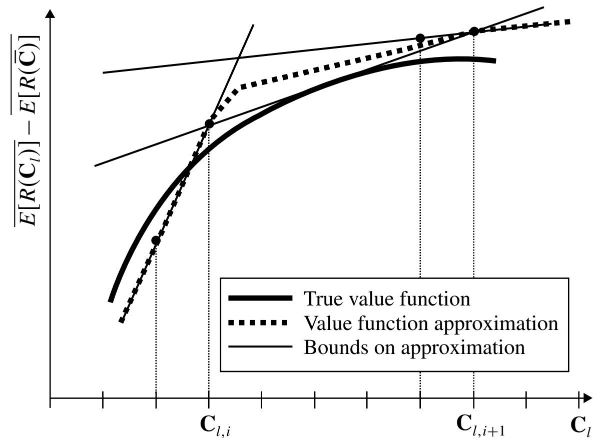

Figure 1. Illustration of the value function interpolation algorithm.

The interpolation algorithm is based on determining how much of the difference between the value function at $\mathbf{C}$ and $\overline{\mathbf{C}}$ can be attributed to the difference in remaining capacity at each of the individual flight legs. For this purpose, we have introduced the auxiliary capacity vectors $\mathbf{C}_l$ that are equal to $\overline{\mathbf{C}}$ except on leg l. When $\mathbf{C}_l$ is between two discretization points in dimension l (Case (1c)), we determine lower and upper bounds on its function value based on the presumed concavity of the value function for a single-leg flight. These bounds are illustrated in Figure 1. The upper bound consists of the lower envelope of the tangent lines at the discretization points, while the lower bound is given by the convex combination of the value function at these two points. We then combine these bounds by taking a weighted average. When $\mathbf{C}_l$ falls outside the range of discretization points (Cases (1a) and (1b)), linear extrapolation is used. Based on this approximation, we determine the difference between the value function at $\mathbf{C}_l$ and $\overline{\mathbf{C}}$, which can be seen as the change of the value function along dimension l. The difference between the value function at $\mathbf{C}$ and $\overline{\mathbf{C}}$ is then estimated by the sum of these changes.

We are now ready to define an algorithm that estimates all value functions recursively. Let $E[R^{T+1}] \equiv 0$. Then, $E[R^t(\mathbf{C}^t)]$ $(t = 1, \ldots, T)$ can be estimated as follows.

Recursive Value Function Estimation Algorithm

Step 1. Use the Discretization Points Selection Algorithm to select discretization points for each time window $t = 1, \ldots, T$.

Step 2. FOR $t = T$ TO 1

(2a) Use the Value Function and First Finite Differences Estimation Algorithm to approximate $E[R^t(\overline{\mathbf{C}}^t)]$, $E[R^t(\overline{\mathbf{C}}^t_{l,i})]$, and $\Delta_l E[R^t(\overline{\mathbf{C}}^t_{l,i})]$ for each discretization point $\overline{\mathbf{C}}^t_{l,i}$.

(2b) Use the Value Function Interpolation Algorithm to approximate $E[R^t(\mathbf{C}^t)]$ for arbitrary $\mathbf{C}^t$ in the state-space.

END FOR

The estimation of the value function might be computationally expensive, but it can be done offline. After $\overline{\mathbf{C}}^t$, the discretization points $\bar{\mathbf{C}}^t_{l,i}$, and the estimates of $E[R^t(\bar{\mathbf{C}}^t)]$, $E[R^t(\bar{\mathbf{C}}^t_{l,i})]$, and $\Delta_l E[R^t(\bar{\mathbf{C}}^t_{l,i})]$ have been calculated and stored, the approximation of $E[R^t(\mathbf{C}^t)]$ reduces to a single call of the Value Function Interpolation Algorithm. It is important, however, that these estimates are updated regularly during the booking process, even for a given set of flight departures. As the airline learns more about the actual realization of the demand process, it should update its demand forecasts for the remaining part of the booking period accordingly. For instance, if the number of advance booking requests for a certain destination was surprisingly high, this may indicate that there is a special event taking place that will lead the number of last-minute booking requests to be higher than usual as well. Ideally, the value function estimates should depend on the number of booking requests received for each *ODF*, but in that case the dimension of the state-space would be too large. Regular updates of the value function estimates may at least help to capture part of this effect. The current approximation based on a stochastic demand model that is fixed throughout the booking process is only intended to adjust the booking policy for the effect of the statistical fluctuations of demand on the remaining capacity, which we believe to be an important factor in effective inventory control.

3.4. The Simulation-Based Booking-Limits Approach

We now can define our proposal to determine a booking-limit policy for an arbitrary time window t.

Simulation-Based Booking-Limit Policy (SBL)

Step 1. (offline, say weekly) Run the Recursive Value Function Estimation Algorithm.

Step 2. (offline, say overnight) Run the Numerical Booking-Limit Improvement Step on top of EMSR-BL or LP-BL (whichever works best) to determine the booking limits $\mathbf{b}^t$. Use the Value Function Interpolation Algorithm to evaluate $E[R^t(\mathbf{C}^t)]$ when necessary.

Step 3. (online) Implement the booking-limit (BL) policy with the set of booking limits $\mathbf{b}^t$.

In the numerical experiments of §4, we have compared both ways to determine a starting point for the Numerical Booking-Limit Improvement Step, which we refer to as SBL_{EMSR} and SBL_{LP}, respectively. For Step 0 of the Value Function and First Finite Differences Estimation Algorithm, we have always used EMSR-BL. Note that the same algorithms can be used with different nesting policies, both standard and theft, which only affects the implementation of the simulation program. For this reason, we have tested the performance of the SBL approach for both methods.

We feel that the most important ideas underlying the SBL method are:

1. The use of Model (1) to determine the nesting order, hence solving the combinatorial aspect of the problem efficiently. Because nesting can be combinatorially explosive, Model (1) provides a feasible, and we think reasonable, approximation to the optimal nesting order.

2. The combination of simulation and approximate dynamic programming to estimate the expected revenue at the end of each time window. This allows capturing the effect of policy updates for the Numerical Booking-Limit Improvement Step.

3. The stochastic gradient algorithm to improve a given set of booking limits.

The SBL approach is clearly a heuristic, because the stochastic gradient algorithm will most likely not have converged to an optimal solution at its termination. Moreover, much of its performance depends on the tuning of the parameters of the subroutines it invokes. However, the algorithm has been designed in such a way that it should always lead to an improvement of the initial set of booking limits, given the simulation model and the accuracy of the value function approximation. The numerical evidence presented in the next section supports this. In contrast, most of the methods that have been proposed in the literature have a better theoretical justification, but they either oversimplify the problem or are directed at optimizing a different policy than the one that is actually implemented.

4. Computational Results

In this section, we conduct computational experiments to determine how much the SBL approach improves over EMSR-BL and LP-BL and what factors affect the relative performance of these policies. We consider some large-scale examples to show that the SBL approach is tractable for realistically sized problems. In addition, we briefly compare the SBL approach with the BP policy for completeness.

4.1. The Simulation Environment

Following Weatherford et al. (1993), for our computational experiments we model the arrival process of booking requests for class *odf* as a nonhomogeneous Poisson process (NHPP) with arrival intensity

$$\lambda_{odf}(t) = \beta_{odf}(t) \times A_{odf}, \tag{5}$$

where $\beta_{odf}(t)$ is the standardized beta distribution—

$$\beta_{odf}(t) = \frac{1}{\tau}\left(\frac{t}{\tau}\right)^{\alpha-1}\left(1-\frac{t}{\tau}\right)^{\beta-1}\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)},$$

in which τ is the length of the booking period, and α and β are parameters defining the arrival pattern—and A_{odf} is a random variable that obeys the gamma distribution. The properties of this so-called *Pólya process* are studied in the monograph by Grandell (1997) on the more general class of mixed Poisson processes. Modeling the rate by the beta distribution $\beta_{odf}(t)$ allows for a wide range of unimodal arrival patterns. The random variable A_{odf}, which has the interpretation of the total demand for booking class *odf*, adds an extra level of randomness to the NHPP. In addition, it introduces a positive correlation between the number of bookings in separate parts of the booking horizon. As a result, the current number of bookings provides information about demand to come, which should be taken into account for the policy updates.

Because the gamma distribution is the conjugate prior of the Poisson distribution, it is easy to show that the total demand D_{odf} generated by the Pólya process has the negative binomial distribution. With a little extra work, it can be shown that the conditional distribution of remaining demand after each update is negative binomial as well. This suggests that the arrival process of booking requests after each update can again be modeled by a Pólya process with arrival intensity (5), where the parameters of the distribution of A_{odf} now depend on the number of booking requests received up to that point. It is understood that, for each policy update, Model (1) and the EMSR method are based on the conditional demand distributions, and that the Numerical Booking-Limit Improvement Step 2 of the SBL approach is based on a conditional model of the arrival process of booking requests. In contrast, as discussed in §3.3.2, the estimate of the value function is based on the "unconditional" prior demand model (5).

Empirical studies by Lyle (1970) and de Boer (1999) suggest that a Pólya process with arrival intensity (5) is a reasonable high-level approximation of airline demand. Note, however, that this model does not incorporate important practical factors such as overbooking, cancellations, and no-shows. In addition, the model assumes that each customer requests a particular fare class and that no other fare class, lower or higher, would do. Although airlines do try to fence off low fares from price-insensitive customers—for instance, by requiring a Saturday night stay—this is unrealistic. More likely, customers are looking for the lowest available fare that meets their restrictions; thus, the RM policy affects demand. In this case, the same control framework (virtual nesting) can be used to determine the seat availability for each booking class, but the booking-limit calculation step would have to be adjusted to take this into account. Because it is hard to model such a demand process in closed form, independent demand for each booking class is a common assumption in the RM literature. However, for the SBL method, this behavior can easily be captured, because only the simulation model would need to be adjusted. For instance, each simulated customer could consecutively request the availability of all fare classes that meet his restrictions, starting with the cheapest one, and would buy the first available. We have deliberately left out these factors in our numerical analysis to concentrate on the effects of including nesting in a probabilistic and dynamic model of demand. These simplifications limit the generality of our results, because we cannot be sure how the SBL method would perform given a more realistic model of the booking process. However, as far as we know, no other method has been proposed to date to address such a sophisticated representation of the problem against which we could benchmark it.

We have implemented a computer program in C++ that simulates the booking process according to the model we outlined. We have implemented the policies EMSR-BL, LP-BL, SBL_{EMSR}, SBL_{LP}, and BP, under both standard and theft nesting, and allowing at most 10 buckets per leg in the clustering step. Unless stated otherwise, different policies are tested on the same simulated sequence of booking requests to get a more accurate estimate of the revenue differential.

4.2. Sensitivity Analysis

We want to identify the factors that affect the relative performance of the considered policies. In particular, we are interested in under what conditions the SBL approach leads to the most significant revenue gains. We consider a single-leg flight and a small network. To isolate these factors from the impact of policy updates and the value function estimation, in the examples below the booking policy is calculated only once at the beginning of the booking period ($T = 1$).

4.2.1. Example 1: Single-Leg Flight. Our first example is a single-leg flight with five fare classes. The factors we consider here are the nominal load factor, the demand variability, and the fare structure. The nominal load factor (LF) is defined as the total expected demand for seats divided by the total number of available seats. Demand variability is measured by the coefficient of variation (CV) of the demand for each booking class. The fare structure is determined by the relative difference (RDF) between two consecutive fares for the same itinerary. For example, an RDF of 50% implies that the difference between the lowest and the highest fare in the market for the same itinerary is roughly a factor of five. In addition, we have examined the effect of the arrival order of booking requests, which is determined by the parameters of the beta distribution. We distinguish between time-homogeneous (HOM) arrivals for each booking class, and the case that the lower-fare classes tend to book early in the booking period, while the higher-fare classes tend to book closer to departure time (LBH).

In Tables 1 and 2 we report the results of 100,000 simulation runs using standard nesting. Here as well as in all other numerical examples, the reported improvement of the SBL method over alternative booking-limit heuristics is using the latter as the starting point for the random search. For completeness, we also report the improvement of EMSR-BL over LP-BL. The most relevant observations from these examples are:

- The gain of SBL over LP-BL increases with demand variability and varies significantly with the fare structure, while the gain of SBL over EMSR-BL is less sensitive to these factors. The reason is that the EMSR method takes into account both demand uncertainty and the fare levels, while the LP-based method does not.

Table 1. Sensitivity gain SBL over EMSR and LP for RDF and CV (standard nesting).

	SBL_{EMSR} − EMSR			SBL_{LP} − LP			EMSR − LP		
RDF ⇓ \ CV ⇒	25%	50%	75%	25%	50%	75%	25%	50%	75%
25%	0.02%	0.05%	0.13%	0.67%	1.62%	2.94%	0.66%	1.56%	2.81%
50%	0.05%	0.11%	0.30%	0.12%	0.73%	1.20%	0.30%	0.61%	0.88%
75%	0.06%	0.11%	0.30%	0.43%	1.11%	1.60%	0.56%	0.99%	1.29%

Table 2. Sensitivity gain SBL over EMSR and LP for LF and arrival process (standard nesting).

	SBL_{EMSR} − EMSR		SBL_{EMSR} − LP		EMSR − LP	
LF ⇓ \ Arrivals ⇒	LBH	HOM	LBH	HOM	LBH	HOM
75%	0.01%	0.06%	0.09%	0.12%	0.08%	0.04%
100%	0.08%	0.48%	0.92%	0.90%	0.85%	0.42%
125%	0.11%	0.68%	0.69%	1.14%	0.59%	0.47%

- The gain of SBL over EMSR-BL and LP-BL generally increases with the load factor, reflecting that a better RM method is more important given heavy demand.
- The gain of SBL over EMSR-BL strongly depends on the arrival process, while the gain of SBL over LP-BL is less sensitive to this factor. The reason is that the EMSR method implicitly assumes that the lowest classes book first, thus the protection levels are off when this is not the case.
- In most cases, the gain of EMSR-BL over LP-BL and the gain of SBL over EMSR-BL add up to roughly the gain of SBL over LP-BL, which suggests that the expected revenue of the SBL approach is relatively insensitive to the starting point of the random search. Note that the resulting policies are not necessarily the same, which may lead to different combinations of load factor and yield.

To investigate the effect of the nesting policy on the relative performance of the SBL approach, in Table 3 we report the result of 100,000 simulation runs using theft nesting. For completeness, we also compare the expected revenue of the SBL approach (EMSR starting point) under standard and theft nesting. The most relevant observations from this example are:

- The gains of the SBL approach over EMSR-BL and LP-BL are much more significant under theft nesting than they were under standard nesting. Unlike the SBL approach, these methods do not take the operational nesting policy into account, while this clearly should affect the booking limits.
- If the SBL approach is used to set the booking limits under both standard and theft nesting, then theft nesting performs better, and more so when the demand variance increases.

Figure 2 shows the estimated objective values of the sequence of booking limits produced by the SBL algorithm with different starting points (EMSR-BL and LP-BL) for the case that CV = RDF = 50%, under both standard and theft nesting. The expected revenue estimates are again based on 100,000 simulation runs. In all cases, the stopping criterion of the Numerical Booking-Limit Improvement Step was met before the maximum number of iterations was reached. Note that the improvement obtained by the SBL Algorithm is not monotone, perhaps because of the discrete and

Figure 2. Convergence of the SBL Algorithm for Example 1 with CV = RDF = 50%.

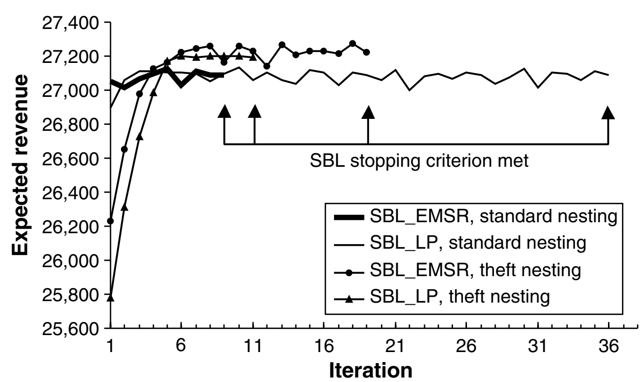

Table 3. Sensitivity gain SBL over EMSR and LP for RDF and CV (theft nesting).

	SBL_{EMSR} − EMSR			SBL_{LP} − LP			SBL_{EMSR} Standard − Theft		
RDF ⇓ \ CV ⇒	25%	50%	75%	25%	50%	75%	25%	50%	75%
25%	7.93%	3.20%	1.80%	11.79%	8.72%	8.00%	−0.15%	−0.24%	−0.78%
50%	7.56%	3.78%	2.34%	8.80%	5.45%	4.03%	−0.35%	−0.40%	−0.65%
75%	6.25%	3.36%	2.45%	7.09%	3.93%	3.06%	−0.13%	−0.45%	−0.89%

Table 4. Sensitivity gain SBL over EMSR and LP for LFR and FLD (standard nesting).

	$SBL_{EMSR} - EMSR$			$SBL_{LP} - LP$			$EMSR - LP$		
FLD ⇓ \LFR ⇒	50%	62.5%	75%	50%	62.5%	75%	50%	62.5%	75%
25%	0.02%	0.50%	1.03%	0.79%	0.99%	1.17%	0.77%	0.49%	0.20%
50%	0.49%	0.70%	0.36%	0.76%	0.81%	1.09%	0.26%	0.14%	0.71%
75%	0.03%	0.07%	0.07%	0.83%	0.67%	0.66%	0.58%	0.60%	0.57%

stochastic nature of the optimization problem, and that most progress is made in the first couple of iterations.

4.2.2. Example 2: Single-Hub Network. We continue our sensitivity analysis with an example of a single-hub network connecting five cities to investigate factors that may affect the performance of the considered policies in a network setting. We only consider one bank of inbound and one bank of outbound flights. The airline offers all 30 possible itineraries, both local and connecting, at a full-fare and a discount rate. The factors that we have considered are the fraction local demand on each leg (FLD) and the average ratio between local and connecting fares (LFR). For example, an LFR of 50% indicates that on average traveling to or from the hub costs half as much as a connecting flight from spoke to spoke. The results of 100,000 simulation runs for an LBH arrival process using standard nesting are reported in Table 4.

The most important observation from this example is that when local demand is relatively low, the gain of SBL over EMSR-BL increases with the value of local traffic, while when local demand makes up a larger fraction of total demand, the effect is unclear. The gain of SBL over LP-BL is less sensitive to these factors. A possible explanation may be that EMSR-BL, by solving the problem for each leg separately, may have a tendency to overprotect local traffic. Note that a connecting passenger should only be turned down when he is expected to displace a local passenger on both of his flight legs. However, leg-based methods based on displacement adjusted or prorated fares set protection levels to avoid displacement on a single leg, which has a higher probability of occurring. This tendency might be stronger when local passengers are relatively more valuable, but this would matter less the larger their share of total traffic. Finally, note that the performance of SBL is generally independent of the starting point of the random search, because again in most cases the gain of EMSR-BL over LP-BL and the gain of SBL over EMSR-BL roughly add up to the gain of SBL over LP-BL.

4.3. Comparison with the Optimal Policy

In our next example, we compare SBL, EMSR-BL, and LP-BL with the optimal booking policy determined by stochastic dynamic programming. To illustrate the importance of the value functions for the performance of the SBL approach, we test an alternative implementation of the Finite Differences Estimation Algorithm that simulates the booking process until the end of the booking period, not accounting for future policy updates (SBL^{NVV}). The comparisons between EMSR-BL and LP-BL and between standard nesting and theft nesting have been left out deliberately, because our primary interest is the relative performance of the SBL approach.

Consider a single-leg flight with capacity of 100 seats. The airline offers six different booking classes, with fares ranging from \$100 to \$800. Booking requests arrive according to a nonhomogeneous Poisson process of the LBH type. The gamma component of the arrival intensity in (5) is suppressed to allow modeling the problem as a one-dimensional Markov Chain ($A_{odf} = E[D_{odf}]$). Average demand exceeds capacity by 30%. The results of 2,500 simulations runs using standard nesting for different numbers of policy updates are reported in Table 5. The reported bounds on the optimality gap are based on the SDP solution (optimal expected revenue of \$26,594.5) and a 95% confidence interval for the true expected revenue of the SBL approach based on the sample average and its standard deviation. The most relevant observations from this table are that:

- The optimality gap of the SBL method decreases with the number of policy updates until, with 95% confidence, it is less than 0.17%.
- The gain of SBL over EMSR-BL and LP-BL increases with the number of policy updates, and when SBL is implemented without accounting for the policy updates, the revenue gains are generally lower. This suggests that it is important to take future updates into account when deter-

Table 5. Comparison with optimal booking control (standard nesting).

Number of updates	0	5	10	20
$SBL_{EMSR} - EMSR$	0.11%	0.23%	0.37%	0.48%
$SBL_{LP} - LP$	0.79%	0.89%	1.07%	1.18%
$SBL^{NVV}_{EMSR} - EMSR$	0.11%	0.16%	0.16%	0.27%
$SBL^{NVV}_{LP} - LP$	0.79%	0.93%	0.86%	1.13%
opt. gap SBL_{EMSR}	0.77%–1.55%	0.11%–0.90%	<0.41%	<0.17%

Table 6. Accuracy of the value function estimates (standard nesting).

Number of updates	0	5	10	20
$\overline{E[R^1(\mathbf{C}^1)]}$	26286.4	26512.8	26548.6	26572.5
$\mathrm{SBL}_{\mathrm{EMSR}}$ revenue	26286.4 (53.3)	26460 (53.8)	26589.4 (53.3)	26649.8 (50.9)

Table 7. Comparison with optimal booking control (theft nesting).

Number of updates	0	5	10	20
$\mathrm{SBL}_{\mathrm{EMSR}} - \mathrm{EMSR}$	4.92%	0.91%	0.40%	0.20%
$\mathrm{SBL}_{\mathrm{LP}} - \mathrm{LP}$	4.97%	1.00%	0.68%	0.89%
$\mathrm{SBL}^{\mathrm{NVV}}_{\mathrm{EMSR}} - \mathrm{EMSR}$	4.92%	0.86%	0.24%	−0.11%
$\mathrm{SBL}^{\mathrm{NVV}}_{\mathrm{LP}} - \mathrm{LP}$	4.97%	0.96%	0.48%	0.50%
opt. gap $\mathrm{SBL}_{\mathrm{EMSR}}$	0.79%–1.57%	<0.79%	<0.33%	<0.09%

mining the current booking policy, which the latter two methods do not.

To illustrate the performance of the Value Function Estimation Algorithm, in Table 6 we report both the average revenue generated by the SBL approach (standard deviation of the *sample average* given between parenthesis) and the estimate of the value function at the beginning of the booking period. Note that the value function estimates are relatively accurate.

In Table 7, we report the results for the same 2,500 realizations of the demand process under theft nesting. The most relevant observations are that:

- The gain of SBL over EMSR-BL and LP-BL now decreases with the number of policy updates. A possible explanation is that theft nesting is more effective when the booking limits are updated more frequently; thus, there is less room for the SBL approach to improve.
- When SBL is implemented without accounting for the policy updates, the revenue gains are lower and can even be negative. We have seen this in many other simulation experiments that we have not reported here as well, which shows that the value function estimation step is in fact essential for the success of the SBL method.
- The revenue gains of the SBL approach are more significant under theft nesting than under standard nesting when the number of policy updates is relatively small, for reasons already explained above.

4.4. Larger Networks

We consider networks of 5, 10, and 15 cities connected by a single hub. The airline offers all possible itineraries, both local and connecting, at five different rates. We consider the case of demand following a pure nonhomogeneous Poisson process ($A_{odf} = E[D_{odf}]$, with $\mathrm{CV} = 1/E[D_{odf}]$, generally less than 30%), and the case of the arrival intensity itself being a gamma-distributed random variable (a Pólya process, with $\mathrm{CV} = 35\%$), under both standard and theft nesting. The other simulation settings are RDF = 50%, LF = 125%, LFR = 75%, FLD = 25%, and an LBH arrival process. The booking policy is updated 20 times during the booking period. The results of 1,000 simulation runs are reported in Table 8. Again, the comparison between standard and theft nesting has been left out deliberately. We applied "slower" SBL optimization settings for the 5-spokes network with demand modeled by a Pólya process than for the other test cases (see §4.4.1), which may be part of the reason why the relative performance of the SBL approach is generally better in this case. Other important observations are:

- The gains of the SBL approach are practically significant. Given the operating scale and cost structure of airlines, seemingly small revenue improvements might translate to millions of dollars each year, added straight to the bottom line.
- The gain of the SBL approach is larger under theft nesting than under standard nesting.
- In all cases, the combined gain of EMSR-BL over LP-BL and of SBL over EMSR-BL is much higher than the gain of SBL over LP-BL, which shows that the EMSR booking limits provide a better starting point for the SBL approach.

4.4.1. Computational Tractability. When the scale of the problem allows it, we can use "slower" optimization

Table 8. Large-scale simulation results (revenue performance).

	$\mathrm{SBL}_{\mathrm{EMSR}} - \mathrm{EMSR}$			$\mathrm{SBL}_{\mathrm{LP}} - \mathrm{LP}$			EMSR − LP		
Test case ⇓ \ # spokes ⇒	5	10	15	5	10	15	5	10	15
Poisson, standard	0.16%	0.09%	0.08%	0.84%	0.24%	0.24%	1.67%	2.07%	2.17%
Poisson, theft	0.18%	0.47%	0.61%	1.23%	0.60%	0.61%	1.50%	1.25%	1.08%
Pólya, standard	0.07%	0.05%	0.05%	1.06%	0.26%	0.25%	1.64%	1.84%	1.85%
Pólya, theft	0.32%	0.38%	0.48%	1.29%	0.84%	0.68%	1.07%	1.35%	1.01%

Table 9. Large-scale simulation results (computational performance).

	VV-EST (hrs)			SIM SBL_{EMSR} (sec)			SIM SBL_{LP} (sec)		
Test case ⇓ \ # spokes ⇒	5	10	15	5	10	15	5	10	15
Poisson, standard	4.84	24.53	49.41	0.33	1.24	14.13	0.45	1.57	12.55
Poisson, theft	4.84	24.53	49.41	0.32	1.19	10.53	0.60	2.97	18.95
Pólya, standard	10.09	42.25	60.52	10.16	3.14	11.30	12.41	3.23	9.73
Pólya, theft	10.09	42.25	60.52	17.98	5.26	9.89	21.07	5.73	14.43

settings for the SBL approach, which may lead to a better solution. The number of iterations (n) for the Finite Differences Estimation Algorithm was 1,000 for the 5-spokes network with demand modeled by a Pólya process, using a fixed stepsize $\rho_k = 0.5$, but only 100 for the other test cases, with stepsize $\rho_k = 0.05$. In all cases, we used $\Delta_{\max} = 2$ and $k_{\max} = 10$. The number of iterations for the Value Function and First Differences Estimation Algorithm was 10,000 for the 5- and 10-spokes networks, but only 5,000 for the 15-spokes network. In all cases, we used $q = 4$ grid points in each dimension of the state-space, while for the Value Function Interpolation Algorithm we used $w = 0.5$ without much tuning. The value function estimate based on standard nesting was used to implement the SBL approach under theft nesting as well, to save computation time. The run time of the value function estimation (VV-EST, in CPU) and of the booking-limit improvement step (approximated by the average simulation time per update of the SBL policy (SIM, in CPU)) for these large-scale examples on a Pentium IV processor are reported in Table 9. The important observations are that:

- The estimation of the value function for the 15-spokes network took about 50 to 60 hours of CPU time, depending on the demand process. Tuning of the optimization parameters and a more efficient implementation of the program can speed up the algorithm, but the important point is that this part of the algorithm can be done offline.
- The online part of the SBL approach takes only seconds for all instances, which shows that the proposed SBL approach is computationally tractable for realistically sized problems.

4.5. Comparison with Bid-Price Control

We now compare the SBL approach with the BP policy defined in §2.2. Clearly, the BP policy is easier to compute, but as we have pointed out, its success may strongly depend on the frequency of policy updates. For this reason, we compare the revenue performance of the SBL approach (EMSR starting point, standard nesting) with 20 updates to the BP policy updated 20 to 100 times. The results of 1,000 simulation runs for the same test cases as in §4.4, but with different sequences of booking requests, are reported in Table 10. Even given 100 updates of the BP policy, the SBL approach with only 20 updates performed best. As we have pointed out, a better implementation of the BP policy may be based on dynamic bid prices, but a more thorough comparison of booking-limit and bid-price control is outside the scope of this research.

4.6. Insights Gained

The most important insights gained from the numerical experiments in this section are:

- The SBL approach can lead to practically significant revenue improvements over both EMSR-BL and LP-BL.
- The SBL approach takes into account factors that should affect the booking-limit policy, but that other methods ignore, such as demand uncertainty and the fare structure (compared to LP-BL), the dynamics of the demand process, the likelihood of a connecting passenger displacing two local travelers, future policy updates, and the operational nesting policy. Because both EMSR-BL and LP-BL implicitly assume standard nesting, the improvement of the SBL approach over these methods is particularly significant under theft nesting.
- The SBL approach is computationally tractable for realistically sized problems.
- The use of value functions to account for future policy updates is essential for the success of the SBL approach.
- EMSR-BL provides a better starting point for the SBL random search than LP-BL.

Table 10. Gain SBL approach with 20 updates over the BP policy ($SBL_{EMSR} - BP$).

	5 spokes		10 spokes		15 spokes	
# updates BP ⇓ \ Demand process ⇒	Pólya	Poisson	Pólya	Poisson	Pólya	Poisson
20	2.19%	1.89%	2.14%	2.67%	2.00%	2.60%
40	1.72%	1.27%	1.74%	2.22%	1.34%	2.01%
60	1.21%	1.21%	1.29%	2.22%	1.15%	1.76%
80	1.27%	1.17%	1.29%	1.97%	1.09%	1.71%
100	1.08%	1.06%	1.33%	2.06%	1.03%	1.59%

5. Conclusions

In this paper, we proposed a framework to address the stochastic and dynamic character of the demand and the nested character of booking-limit control in a network environment. Starting with any nested booking-limit policy, we combine a stochastic gradient algorithm and approximate dynamic programming ideas to improve the initial booking limits. Preliminary simulation experiments suggest that this approach (a) is computationally feasible for realistically sized networks, because the computationally demanding part of the algorithm can be done offline, and (b) has the potential of leading to practically significant revenue enhancements over policies based on inaccurate representations of booking-limit control in a network. The simplified demand model and the relatively small number of test problems limit the extent of our conclusions, but the potential revenue gains warrant more extensive testing.

Appendix

Clustering Algorithm

On each leg l, recursively cluster all booking classes with nonnegative adjusted leg revenue into at most $N_{\max}$ (to be chosen beforehand) leg buckets as follows. Let $k=0$ be the number of buckets B_i^l $(1\leqslant i\leqslant k)$ already defined, and consider the range of adjusted leg revenues of all booking classes that remain to be clustered into at most $N_{\max}-k$ additional buckets. This range starts at 0 and ends at

$$UB_{k+1}^l=\max\left\{\bar{f}_{odf}^l\geqslant 0 \mid odf\in S_l\setminus\bigcup_{i=1}^k B_i^l\right\}.$$

Divide this range into $N_{\max}-k$ equal parts and cluster all booking classes whose adjusted leg revenue falls into the last subinterval into bucket B_{k+1}^l, thus

$$B_{k+1}^l=\left\{odf\in S_l\colon\ \frac{N_{\max}-k-1}{N_{\max}-k}UB_{k+1}^l\leqslant\bar{f}_{odf}^l\leqslant UB_{k+1}^l\right\},$$

which by definition contains the booking class with the highest adjusted leg revenue that had not yet been clustered. Let $k\leftarrow k+1$ and terminate the recursion when $k=N_{\max}$, or when no more booking classes with nonnegative adjusted leg revenue remain to be clustered. Finally, let $N_l=k+1$ be the total number of buckets on leg l after clustering all booking classes with negative adjusted leg revenue into the lowest bucket

$$B_{N_l}^l=\{odf\in S_l\colon\ \bar{f}_{odf}^l<0\}$$

if this set is not empty. Otherwise, $N_l=k$. This basic algorithm, which may be different from what airlines use in practice, guarantees that there are no empty buckets and reduces the number of buckets on a leg if the adjusted leg revenues are closely grouped together.

LP-Based Booking-Limit Calculation Method (LP-BL, Williamson 1992)

Let $b_1^l=C_l$. Let p_i^l be the number of seats that needs to be protected (*protection level*) on leg l for buckets 1 to i from bookings for bucket $i+1$ or lower, which based on the LP solution is

$$p_i^l=\sum_{j=1}^i\sum_{odf\in B_j^l}x_{odf}^*\quad(i=1,\ldots,N_l-1).$$

Then, it makes intuitive sense to let

$$b_{i+1}^l=C_l-p_i^l\quad(i=1,\ldots,N_l-1).\tag{6}$$

Note that $b_i^l\geqslant 0$ for all i because of the capacity constraints of Model (1).

EMSRb Booking-Limit Calculation Method (EMSR-BL, Belobaba 1992)

Let D_i^l be the aggregate demand for all booking classes in bucket i on leg l, and let r_i^l be a weighted average of their adjusted leg revenues; thus,

$$D_i^l=\sum_{odf\in B_i^l}D_{odf},$$

$$r_i^l=\frac{1}{E[D_i^l]}\sum_{odf\in B_i^l}E[D_{odf}]\bar{f}_{odf}^l.$$

Let $D_{1:i}^l$ be the aggregate demand for buckets 1 to i; thus,

$$D_{1:i}^l=\sum_{j=1}^i D_j^l=\sum_{j=1}^i\sum_{odf\in B_j^l}D_{odf}.$$

We approximate the distribution of $D_{1:i}^l$ with the Gaussian distribution with

$$E[D_{1:i}^l]=\sum_{j=1}^i\sum_{odf\in B_j^l}E[D_{odf}],$$

$$\mathrm{Var}[D_{1:i}^l]=\sum_{j=1}^i\sum_{odf\in B_j^l}\mathrm{Var}[D_{odf}].$$

The average weighted leg revenue $r_{1:i}^l$ for these buckets is

$$r_{1:i}^l=\frac{1}{E[D_{1:i}^l]}\sum_{j=1}^i E[D_j^l]r_j^l.$$

We can now define the booking limits. As before, let $b_1^l=C_l$. For $i=2,\ldots,N_l-2$, the number of seats that needs to be protected for buckets 1 to i from bookings from bucket $i+1$ (and thus from all lower buckets) is

$$p_i^l=\max\{p\mid r_{1,i}^l P(D_{1,i}^l\geqslant p)>r_{i+1}^l,\ p\ \text{ integer}\},$$

and the booking limit corresponding to bucket $i+1$ is again given by (6). Finally, set $b_{N_l}^l=0$, because we do not want to accept bookings that seem to have a negative contribution to leg revenue.

Acknowledgments

The authors thank Garrett van Ryzin and the three reviewers of this paper for numerous comments and suggestions that improved it substantially. This research was partially supported by the MIT-Singapore Alliance.

References

Belobaba, P. P. 1987. Air travel demand and airline seat inventory management. Ph.D. thesis, Flight Transportation Laboratory, Massachusetts Institute of Technology, Cambridge, MA.

Belobaba, P. P. 1989. Application of a probabilistic decision model to airline seat inventory control. *Oper. Res.* **37** 183–197.

Belobaba, P. P. 1992. Optimal vs. heuristic methods for nested seat allocation. *Proc. AGIFORS Reservations and Yield Management Study Group.* Brussels, Belgium.

Belobaba, P. P., L. R. Weatherford. 1996. Comparing decision rules that incorporate customer diversion in perishable asset revenue management situations. *Decision Sci.* **27** 343–363.

Bertsekas, D. P. 1995. *Dynamic Programming and Optimal Control*. Athena Scientific, Belmont, MA.

Bertsimas, D., I. Popescu. 2000. Revenue management in a dynamic network environment. Working paper, Sloan School of Management, Massachusetts Institute of Technology, Cambridge, MA.

Bratu, S. 1999. Network value concept in airline revenue management. Master's thesis, Operations Research Center, Massachusetts Institute of Technology, Cambridge, MA.

Brumelle, S. L., J. I. McGill. 1993. Airline seat allocation with multiple nested fare classes. *Oper. Res.* **41** 127–137.

Brumelle, S. L., J. I. McGill, T. H. Oum, K. Sawaki, M. W. Tretheway. 1990. Allocation of airline seats between stochastically dependent demands. *Transportation Sci.* **24** 183–192.

Chatwin, R. E. 1996. Multi-period airline overbooking with multiple fare classes. *Naval Res. Logist.* **43** 603–612.

Chatwin, R. E. 1998. Multi-period airline overbooking with a single fare class. *Oper. Res.* **46** 805–819.

Chen, V. C. P., D. Günther, E. L. Johnson. 1998. A Markov-decision problem based approach to the airline YM problem. Working paper, The Logistics Institute, Georgia Institute of Technology, Atlanta, GA.

Chen, V. C. P., D. Günther, E. L. Johnson. 1999. Airline yield management: Optimal bid prices for single-hub problems without cancellations. Working paper, The Logistics Institute, Georgia Institute of Technology, Atlanta, GA.

Cross, R. G. 1997. *Revenue Management. Hardcore Tactics for Market Domination*. Broadway Books, New York.

Curry, R. E. 1990. Optimal airline seat allocation with fare classes nested by origins and destinations. *Transportation Sci.* **24** 193–204.

de Boer, S. V. 1999. Mathematical programming in airline seat inventory control. Master's thesis, Department of Mathematics, Vrije Universiteit, Amsterdam, The Netherlands.

de Boer, S. V., R. Freling, N. Piersma. 2002. Mathematical programming for network revenue management revisited. *Eur. J. Oper. Res.* **137** 72–92.

Ermoliev, Yu. 1988. Stochastic quasigradient methods. Yu. Ermoliev, R. J-B. Wets, eds. *Numerical Techniques for Stochastic Optimization*. Springer-Verlag, New York, 141–185.

Gaivoronski, A. 1988. Stochastic quasigradient methods and their implementation. Yu. Ermoliev, R. J-B. Wets, eds. *Numerical Techniques for Stochastic Optimization*. Springer-Verlag, New York, 313–351.

Gallego, G., G. van Ryzin. 1997. A multi-product dynamic pricing problem and its applications to network yield management. *Oper. Res.* **45** 24–41.

Glover, F., R. Glover, J. Lorenzo, C. McMillan. 1982. The passenger mix problem in the scheduled airlines. *Interfaces* **12** 73–79.

Grandell, J. 1997. *Mixed Poisson Processes*. Chapman & Hall, London, U.K.

Karaesmen, I., G. J. van Ryzin. 1998. Overbooking with substitutable inventory classes. Working paper, Graduate School of Business, Columbia University, New York.

Lautenbacher, C. J., S. J. Stidham. 1999. The underlying Markov decision process in the single-leg airline yield management problem. *Transportation Sci.* **33** 136–146.

Lee, T. C., M. Hersh. 1993. A model for dynamic airline seat inventory control with multiple seat bookings. *Transportation Sci.* **27** 252–265.

Littlewood, K. 1972. Forecasting and control of passenger bookings. *AGIFORS Sympos. Proc.*, Vol. 12. Nathanya, Israel.

Lyle, C. 1970. A statistical analysis of the variability in aircraft occupancy. *AGIFORS Sympos. Proc.*, Vol. 10. Terrigal, Australia.

McGill, J. I., G. J. van Ryzin. 1999. Revenue management: Research overview and prospects. *Transportation Sci.* **33** 233–256.

Pagé, J. F. 1999. Experience with O/D management at Air Canada. Presented to AGIFORS Reservations and Yield Management Study Group, London, U.K.

Robinson, L. W. 1995. Optimal and approximate control policies for airline booking with sequential nonmonotonic fare classes. *Oper. Res.* **43** 252–263.

Saranathan, K., K. Peters, M. Towns. 1999. Revenue management at United Airlines. Presented to AGIFORS Reservations and Yield Management Study Group, London, U.K.

Simpson, R. W. 1989. Using network flow techniques to find shadow prices for market and seat inventory control. Memorandum M89-1, Flight Transportation Laboratory, Massachusetts Institute of Technology, Cambridge, MA.

Smith, B. C., C. W. Penn. 1988. Analysis of alternative origin-destination control strategies. *AGIFORS Sympos. Proc.*, Vol. 28. New Seabury, MA.

Subramanian, J., C. J. Lautenbacher, S. J. Stidham. 1999. Yield management with overbooking, cancellations and no shows. *Transportation Sci.* **33** 147–167.

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

Talluri, K. T., G. J. van Ryzin. 1999. A randomized linear programming method for computing network bid prices. *Transportation Sci.* **33** 207–216.

van Ryzin, G. J. 1998. What is nesting on a network. Presented to AGIFORS Reservations and Yield Management Study Group, Melbourne, Australia.

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

Weatherford, L. R. 1998. A tutorial on optimization in the context of perishable-asset revenue management for the airline industry. G. Yu, ed. *Operations Research in the Airline Industry.* Kluwer Academic Publishers, Amsterdam, The Netherlands, 68–100.

Weatherford, L. R., S. E. Bodily, P. E. Pfeifer. 1993. Modeling the customer arrival process and comparing decision rules in perishable asset revenue management situations. *Transportation Sci.* **27** 239–251.

Williamson, E. L. 1992. Airline network seat inventory control: Methodologies and revenue impacts. Ph.D. thesis, Flight Transportation Laboratory, Massachusetts Institute of Technology, Cambridge, MA.

Wollmer, R. D. 1986. A hub-spoke seat management model. Unpublished company report, Douglas Aircraft Company, McDonnell Douglas Corporation, Long Beach, CA.

Wollmer, R. D. 1992. An airline seat management model for a single leg route when lower fare classes book first. *Oper. Res.* **40** 26–37.