OPERATIONS RESEARCH
Vol. 52, No. 1, January–February 2004, pp. 83–104
ISSN 0030-364X | EISSN 1526-5463 | 04 | 5201 | 0083

DOI 10.1287/opre.1030.0079

Overbooking with Substitutable Inventory Classes

Itir Karaesmen
Robert H. Smith School of Business, 4357 Van Munching Hall, University of Maryland, College Park, Maryland 20742, itir@umd.edu

Garrett van Ryzin
412 Uris Hall, Columbia University, New York, New York 10027, gjv1@columbia.edu

This paper considers an overbooking problem with multiple reservation and inventory classes, in which the multiple inventory classes may be used as substitutes to satisfy the demand of a given reservation class (perhaps at a cost). The problem is to jointly determine overbooking levels for the reservation classes, taking into account the substitution options. Such problems arise in a variety of revenue management contexts, including multicabin aircraft, back-to-back scheduled flights on the same leg, hotels with multiple room types, and mixed-vehicle car rental fleets. We model this problem as a two-period optimization problem. In the first period, reservations are accepted given only probabilistic knowledge of cancellations. In the second period, cancellations are realized and surviving customers are assigned to the various inventory classes to maximize the net benefit of assignments (e.g., minimize penalties). For this formulation, we show that the expected revenue function is submodular in the overbooking levels, which implies the natural property that the optimal overbooking level in one reservation class decreases with the number of reservations held in the other reservation classes. We then propose a stochastic gradient algorithm to find the joint optimal overbooking levels. We compare the decisions of the model to those produced by more naive heuristics on some examples motivated by airline applications. The results show that accounting for substitution when setting overbooking levels has a small, but still significant, impact on revenues and costs.

Subject classifications: Transportation: overbooking, yield management. Inventory/production: perishable inventory, stochastic. Probability: stochastic optimization.
Area of review: Transportation.
History: Received October 1998; revisions received January 2001, December 2002; accepted January 2003.

1. Introduction

The idea of yield management in general is to improve revenues by more effectively managing the pricing and allocation of service capacity. See Belobaba (1989), Kimes (1989), and Weatherford and Bodily (1992) for general introductions to yield management, and McGill and van Ryzin (1999) for a recent research survey. Overbooking—that is, accepting more reservations than one has physical capacity to serve as a hedge against cancellations and no-shows—is one of the oldest and, from a revenue standpoint, most important of yield management tactics. Indeed, Smith et al. (1992) at American Airlines estimate that 15% of seats on sold-out flights would be lost if overbooking were not practiced and that the benefit of overbooking at American in 1990 exceeded $225 million. Rothstein (1985) provides an excellent and very readable account of the history of overbooking in the airline industry. Early analyses of the problem in the literature are due to Taylor (1962), Thompson (1961), Rothstein and Stone (1967), and Rothstein (1971, 1974). See also Alstrup et al. (1986), Bitran and Gilbert (1996), Chatwin (1992, 1999), Subramanian et al. (1999), Liberman and Yechiali (1978), and Shlifer and Vardi (1975).

Most of these past works have considered only a single "type" of inventory (e.g., coach cabin seats or standard hotel rooms). Some exceptions are Thompson's (1961) early work, which analyzed the distribution of oversales for two cabins (first/coach) with a limited substitution structure and Ladany's (1976) work on overbooking of hotel customers that demand single or double rooms. Ladany (1976) used restricted policies to allocate the rooms to customers and developed a two-dimensional dynamic programming model to solve the problem. Alstrup et al. (1986) also proposed a dynamic programming formulation of a problem with two cabins, in which the terminal conditions allow for upgrading and downgrading. Interestingly, Rothstein (1971) noted that the (now defunct) Civil Aviation Board at that time expressly prohibited airlines from basing reservation policies on the possibility of reassigning passengers to a different cabin. Also, Rothstein (1985) notes that the International Air Transport Association (IATA) at one time forbade the practice of putting a coach passenger in the first-class cabin without extra charges.

However, there are many examples of real-world operations that involve overbooking multiple inventory classes that can substitute for one another. At most airlines, overbooked business and economy-class passengers are indeed upgraded to the first-class cabin on an oversold flight. Similarly, hotel customers are upgraded to luxury rooms; mid-size cars may be substituted for compact cars, etc., depending on the realized demand and available capacity. See Geraghty and Johnson (1996), Carrol and Grimes (1995), and Edelstein and Melnyk (1977) for work on car rental yield management problems. A less obvious, but important, example occurs in airlines that have frequent departures on the same route, in which case later departures can be used to service the overflow from earlier, overbooked flights (see Ratliff 1998). Of course, inventory classes are usually not perfect substitutes; a van may not be a good substitute for a sports car, a noon flight may be a poor substitute for a morning flight, and so on. Because substitution may decrease (or increase) the quality of service, it has an important impact on customers. At the same time, substitution is a potentially valuable option to enhance the effectiveness of overbooking. Therefore, it is important to understand how to balance its potential benefits and costs.

In this paper, we consider an overbooking problem in which we can accept or deny requests for n different reservation classes. Demands for reservation classes can be satisfied using any one of m different classes of inventory. The cost (real and/or goodwill cost) of assigning demand to an inventory class depends on the particular reservation-inventory class pairing. In our model of the booking process, there are two periods: (1) a reservation period, in which reservations are accepted; and (2) a service period, in which demand is realized and assigned to the m inventory classes. Here, "service period" is used to denote the time interval during which customers with reservations actually show up and must be served or turned away. It is the time of flight departure, the time when customers check in at a hotel, or the time when customers pick up their rental cars. Given the various substitution options, substitution costs, and a probabilistic model of realized demand as a function of the number of reservations on hand, we analyze the problem of setting optimal joint overbooking levels for the n reservation classes to maximize revenues net of penalties.

In reality, acceptances and cancellations tend to take place sequentially over time in most industries. However, it is common practice to solve a simpler, static problem periodically as a heuristic for the true sequential problem. See, for example, Taylor (1962), Thompson (1961), and Rothstein and Stone (1967). This is the approach we adopt here. Our two-stage model is the natural multiclass extension of these traditional static overbooking models.

Several researchers have addressed dynamic models of overbooking. Chatwin (1992, 1999), Rothstein (1971, 1974), and Subramanian et al. (1999), for example, analyze single-class, dynamic programming models of overbooking. Alstrup et al. (1986) uses a dynamic programming model for a sequential problem with two inventory classes that allows for upgrading and downgrading among the inventory classes. However, their approach is computationally intensive and the authors had to aggregate the state space to make it practical. With more than two inventory classes or more general substitution structure, the rapidly increasing size of the state space makes an exact dynamic programming approach intractable. Our two-stage model, in contrast, is solvable for very large numbers of classes and very general substitution structures.

The basic mechanics of our model also combine elements of inventory/production models with random yields with those of models of substitutable products. The random yield in inventory theory is analogous to the number of reservations remaining after the cancellations in our case. Yano and Lee (1995) provide a good review of this literature. Bitran and Gilbert (1994) and Chen (1997) analyze inventory models with substitutable resources with deterministic yields. The facility location model in Jones et al. (1995) and the multilocation inventory problem in Robinson (1990) can be regarded as generalization of substitutable inventory problems, again without incorporating random yield. Bitran and Dasu (1992), on the other hand, combine substitution with random yields and present an infinite horizon stochastic programming formulation. They assume that the number of yield outcomes at each stage is finite and propose two approximation algorithms for the problem: a rolling horizon procedure which is further simplified by assuming yield becomes deterministic after a few periods and a heuristic to allocate items to customers. Hsu and Bassok (1999) also work with random yields and substitutable products. They model a single-period, multiple-product decision problem that allows "downward" substitution as a two-stage stochastic program. They decompose their allocation problem into a parameterized network problem. They propose different solution methods and discuss computational issues. In these last two papers, the authors assume the yield is a random multiple of the production quantity, which allows them to use stochastic programming.

The remainder of this paper is organized as follows: The model and formulation are given in §2. In §3.1, we analyze properties of the deterministic (ex post) service period allocation problem, the most important of which are that its value is *submodular* and *componentwise concave* with respect to the realized demand levels. This means that an increase in the realized demand of one class reduces the marginal benefit of additional demand in any other class. In §3.2, we extend these properties to the expected net revenue function of the overbooking problem in the reservation period using stochastic convexity ideas. Specifically, we show that the expected net revenue function is componentwise concave in each booking level and that the marginal benefit of accepting an additional booking in any class i is nonincreasing in the level of bookings for all other reservation classes. This means that optimal booking levels for class i will decline as the number of reservations on hand for the other classes increases. This property is quite natural, because reservations indirectly compete for the same capacity due to substitution. It also shows qualitatively how reservation levels of one class should affect the optimal overbooking levels of other classes. Finally, in §4, we develop a simulation-based optimization method for determining joint overbooking levels. The algorithm is applied to several examples in §5 to determine what effect the substitution option has on revenue performance and service levels. Our conclusions are given in §6.

2. Model and Formulation

As mentioned, we model the overbooking problem as a two-period problem, consisting of a *reservation period* followed by a *service period*. At the start of the reservation period, we assume that for each reservation class i, there are x_i reservations currently on hand. The problem is to decide how many additional reservation requests to accept. The decision variables, u_i, $i = 1, \ldots, n$, are the number of class i reservations to hold at the end of the reservation period, which we call the *overbooking levels*. Disposing of reservations on hand is not allowed, so $u_i \geqslant x_i$ for all $i = 1, \ldots, n$. In our theoretical analysis, we assume the reservation demand is sufficient to allow any set of overbooking levels u to be chosen. That is, the overbooking levels are not constrained by future demand. However, in our numerical algorithm and testing, we show how limited future demand can be incorporated (at least heuristically) into the model, either by using explicit bounds on the decision variables u or by an appropriate modification of our simulation-based optimization method.

Following the reservation period, cancellations and no-shows are realized, and all remaining customers are either assigned to an inventory class or are denied service. This assignment of customers to reservation classes is modeled as a network flow problem, which is described further in §2.2.

Let the random variable Z_i be the number of customers from reservation class i that actually show up for service. These are the accepted customers who *survive* from the reservation period to the service period. The number of survivals Z_i is a function of the overbooking level u_i, so we denote $Z_i = Z_i(u_i)$. Therefore, the number of cancellations and no-shows is $u_i - Z_i(u_i)$. Henceforth, we use vector notation whenever we refer to all reservation classes, such as x for $(x_1, \ldots, x_n)$, u for $(u_1, \ldots, u_n)$, and $Z(u)$ for $(Z_1(u_1), \ldots, Z_n(u_n))$. We do not distinguish between cancellations and no-shows in our model; the fact that a customer does not survive to the service period could be due to either one.

The revenue gained by accepting a reservation of class i is denoted r_i. We allow for the fact that cancellations may be partially refundable and let the refund associated with a cancellation of reservation class i be denoted q_i. We assume $q_i \leqslant r_i$. We let r and q denote the vector of revenues and refunds, respectively. Let $V_0(z, c)$ denote the value of the service period allocation when the vector of inventory capacities is c and the vector of number of customers who survive is z. This function is defined and analyzed in §2.2. Finally, let $G(u)$ be the expected value of future revenues and costs (*net revenue*) as a function of the overbooking levels, u. (In this formulation, revenues due to the reservations already on hand, x, are considered sunk.)

The single-period problem is then,

$$\max_{u \geqslant x} G(u), \tag{1}$$

where

$$G(u) = \sum_{i=1}^{n} r_i(u_i - x_i) - E\left[\sum_{i=1}^{n} q_i(u_i - Z_i(u_i))\right] + E[V_0(Z(u), c)] \tag{2}$$

and the expectation above is with respect to the random vector of survivals $Z(u)$. This model can be used sequentially over time by updating the problem parameters as they change or as more information becomes available, to update the overbooking levels u.

To understand the overbooking levels produced by this model, we need to analyze the properties of the objective function. The first term in $G(u)$ is linear in u. The second term, however, depends on the probabilistic model of survivals. The third term is even more complicated, depending on both the probabilistic model of survivals and the properties of the service period value function V_0.

2.1. Cancellation Models

We assume that the probability of cancellation over a given period is independent of the time the reservation is made. (This memoryless property has been tested empirically in the airline industry; see Martinez and Sanchez 1970.) Thus, the number of surviving reservations is only a function of the overbooking level u at the end of the reservation period. Further, we will assume that the random variables $\{Z_i(u_i) \mid u_i \geqslant 0\}$ have the *semigroup property*: If Y_1 and Y_2 are independent and Y_1 and Y_2 are stochastically equivalent to (have the same probability distribution as) $Z_i(u_i)$ and $Z_i(s_i)$, respectively, then $Y_1 + Y_2$ is stochastically equivalent to $Z_i(u_i + s_i)$. Finally, we also assume $Z_i(u_i), i = 1, \ldots, n$, are mutually independent and nonnegative.

One model that satisfies these assumptions—and is commonly used in traditional overbooking models—is the binomial model. In the binomial model, let p_i denote the probability of survival for reservation class i over a given period of time (usually the current time until the end of the horizon). Then, the total number of survivals, $Z_i(u_i)$, is binomial distributed with parameters p_i and u_i where u_i is a nonnegative integer. We will also consider a Poisson model as a continuous approximation to the binomial. This model assumes $Z_i(u_i)$ is Poisson with mean $u_i p_i$. We are assuming that the probability of survival for a reservation, p_i, is independent of u_i. Thus, both the binomial and the Poisson models satisfy the semigroup property (see Shaked and Shanthikumar 1988).

2.2. The Service Period

The service period is modeled as a deterministic allocation problem, in which surviving customers are allocated to inventory classes to maximize the total net benefit. The following notation is used in formulating this deterministic problem:

a_{ij}: The net benefit of assigning a customer of reservation class i to inventory class j during the service period (objective function coefficients).

c_j: The capacity of inventory class j.

z_i: The number of customers of reservation class i that show up at the service period (number of survivals).

y_{ij}: The number of customers of reservation class i assigned to inventory class j during the service period (decision variables).

The objective function coefficients, a_{ij}, take into account any real costs and loss of goodwill incurred in case of downgrading and any real revenues or gain in goodwill received in case of upgrading the service required. Note that this formulation allows for a general substitution structure, where any inventory class can provide service (at a cost/benefit) to any reservation class. (This can be modified as discussed below to allow for various restrictions on the assignments.) Fundamentally, however, ours is a cost-based model, and we cannot directly model service-level constraints (e.g., percent of customers denied service less than a given threshold).

We add a virtual inventory class, class $j = 0$, to account for denied service. This class has finite but very high capacity, and assigning a customer to that class means that the customer is denied service. Assuming high capacity for inventory class 0 and assuming nonnegativity for z and for c ensures feasibility of the problem. The assignment variables corresponding to the virtual class are y_{i0}, and the objective function coefficients, a_{i0}, take into account the loss of goodwill cost incurred by denying service to customers of reservation class i, as well as any other direct compensation costs.

Let z denote the n-vector of the number of surviving customers and c denote the $(m+1)$-vector of inventory class capacities (including the denied-service, virtual-class capacity, c_0). The maximum value obtained during the service period is denoted by $V_0(z, c)$. It should be kept in mind that the vector z is actually a realization of the vector of random variables $Z(u)$. Although the ex post service allocation problem is analyzed here in isolation, the results ultimately have important implications for the booking levels in the ex ante reservation period problem.

The allocation problem can be represented as:

$$\text{(TP)} \quad V_0(z, c) = \text{Max} \sum_{i=1}^{n} \sum_{j=0}^{m} a_{ij} y_{ij}$$

$$\text{subject to}$$

$$\sum_{j=0}^{m} y_{ij} = z_i, \quad i = 1, \ldots, n, \tag{3}$$

$$\sum_{i=1}^{n} y_{ij} \leqslant c_j, \quad j = 0, 1, \ldots, m, \tag{4}$$

$$y_{ij} \geqslant 0, \quad i = 1, \ldots, n, \quad j = 0, 1, \ldots, m.$$

This is a transportation problem (TP) in which the supplies are the available inventories, demands are the customers requesting service, and we are maximizing the objective function rather than minimizing. The total supply exceeds the total demand in our formulation.

Let the dual variables associated with constraints (3) and (4) in TP be $\mu = (\mu_1, \ldots, \mu_n)$ and $\lambda = (\lambda_0, \ldots, \lambda_m)$, respectively. The dual of the transportation problem is

$$\text{(DTP)} \quad \text{Min} \sum_{i=1}^{n} z_i \mu_i + \sum_{j=0}^{m} c_j \lambda_j$$

$$\text{subject to}$$

$$\mu_i + \lambda_j \geqslant a_{ij}, \quad i = 1, \ldots, n, \; j = 0, 1, \ldots, m, \tag{5}$$

$$\lambda_j \geqslant 0, \quad j = 0, 1, \ldots, m.$$

Looking at the constraint sets of TP and DTP, we see that both problems have feasible solutions for $z \geqslant 0$. Therefore, the optimal objective function values are equal and finite. Notice that the feasible region of DTP is independent of z and c; it only depends on the network structure and the revenues and costs of the service allocation problem.

The network model can also be extended to allow for piecewise and linear, increasing, and convex overbooking costs. This is handled by simply adding additional virtual inventory nodes with upper bounds on capacity that correspond to the break points of the cost function, and with arc costs that equal the increasing marginal costs of overbooking. Customers that are denied service in the TP are then assigned to the lowest-cost virtual classes first up to the capacity limits, producing the desired increasing and convex cost structure.

Note that by using this network formulation, we are making the assumption that the service provider can make a joint allocation decision with perfect knowledge of the number of survivals in each class. Needless to say, this should be considered only an approximation of reality. While is some cases assignments are made with near-perfect knowledge (e.g., airlines managing two cabins of service in which upgrades are used to handle oversales), more often the assignment process itself is sequential and must be made with imperfect information about the total number of survivals of each class. Indeed, Bitran and Gilbert (1996) specifically focus on the sequential subproblem of "walking" early-arriving customers prior to observing the number of guaranteed, late-arriving customers, the latter being more costly and difficult to relocate. However, we view the network model in our formulation as simply a means of *approximating* the cost structure of the service period—not as a method for making service period decisions directly. As a cost approximation, the network model has the advantage of being computationally efficient, yet capable of capturing the essential substitution costs in a quite general framework. It therefore represents, we believe, a reasonable trade-off between realism and tractability.

3. Structural Properties

In this section, we study properties of the TP, which is a deterministic problem, and provide the extension of those properties to the expected value of the service period value function. We primarily show the natural relationship between overbooking levels of different reservation classes by establishing the submodularity property of the service period value function. The following definitions and theorem—taken from Sundaram (1996)—are needed to establish the result.

Definition 1 (decreasing differences). Let $S \subset R^n$. For $s \in S$, we denote (s_{-ij}, s'_i, s'_j) the vector s, with s_i and s_j replaced by s'_i and s'_j, respectively. A function $f\colon S \to R$ is said to satisfy *decreasing differences* on S if for all $s \in S$, for all distinct i and j in $\{1, \ldots, n\}$, and for all s'_i and s'_j such that $s'_i \geqslant s_i$ and $s'_j \geqslant s_j$, the following relation holds:

$$f(s_{-ij}, s'_i, s'_j) - f(s_{-ij}, s'_i, s_j) \leqslant f(s_{-ij}, s_i, s'_j) - f(s_{-ij}, s_i, s_j).$$

Theorem 1. *A function $f\colon S \subset R^n \to R$ is submodular on S if and only if f has decreasing differences on S.*

Definition 2. A function $f\colon S \subset R^n \to R$ is *supermodular* on S if $-f$ is submodular.

3.1. The Service Period Value Function

Our structural results ultimately derive from the network structure of the TP. Samuelson (1952) showed that the optimal dual solution to a transportation problem is monotonic with respect to the supply capacities. Shapley (1962) showed a similar result for the assignment problem: The optimal objective function of an assignment problem has increasing differences with respect to any two supply nodes or two demand nodes, and it has decreasing differences with respect to a supply node and a demand node. Shapley (1962) stated that the same result can be generalized to a transportation problem. Erlenkotter (1970) showed the monotonicity of dual variables with respect to supply and demand in the transportation problem.

An extension of these properties is the submodularity (supermodularity) of the service period value function (objective function of a transportation problem) with respect to the number of reservations (demand). We provide that result without a proof. It follows from Theorem 3.4.1 of Topkis (1998). In our formulation, TP and DTP are both feasible, so the optimal solution exists for both problems. The set of feasible solutions (μ, λ) for DTP is a sublattice of R^{n+m} and does not depend on (z, c). Hence, all properties required by the proof in Theorem 3.4.1 of Topkis (1998) are satisfied. An alternative proof which uses monotonicity of dual variables and is based on Erlenkotter's (1970) proof is presented in Karaesmen (2001).

Lemma 1. *Function $V_0(z, c)$ is submodular with respect to $(z_1, \ldots, z_n)$.*

The submodularity property is intuitive. It says that the marginal benefit of an additional customer of a certain booking class is nonincreasing in the number of customers that show up from other classes when capacity for inventory classes is fixed. Likewise, the marginal benefit of an additional inventory unit in one class is nonincreasing in the number of units available in other inventory classes. This is because, under substitution, class k customers compete with class l customers for the same scarce capacity.

The next structural property, joint concavity, is elementary and follows from standard linear programming theory.

Lemma 2. *$V_0(z, c)$ is jointly concave in $z_1, \ldots, z_n$ and in $c_1, \ldots, c_m$.*

Both submodularity and concavity hold for other substitution structures, as long as there is a bipartite graph underlying the allocation problem and there is excess capacity in the virtual inventory class (which ensures feasibility of TP and DTP). For instance, if we allowed only upgrading but no downgrading, then we would still have a submodular and concave net revenue function. One can also assign very high overbooking costs to certain reservation classes and penalize their assignment to class 0 if overbooking for those classes is not desirable.

Because capacity is typically fixed, we henceforth drop the capacity c from our notation and denote the service period net revenue function by $V_0(z)$, where z is the vector of number of surviving reservations.

3.2. Properties of the Expected Value Function

We next show that the properties of componentwise concavity and submodularity extend to the expected profit function $G(u)$ provided the random number of survivals $Z_i(u_i)$ have the semigroup property, such as the binomial or the Poisson models (see §2.1 for earlier discussion).

Theorem 2. *For each $i = 1, \ldots, n$, the nonnegative random variable $\{Z_i(u_i) \mid u_i \geqslant 0\}$ has the semigroup property with respect to the parameter u_i, then the function, $G(u)$, defined by (2) is componentwise concave in each u_i, $i = 1, \ldots, n$, and submodular in $(u_1, u_2, \ldots, u_n)$.*

Proof. We prove the result for the case $n = 2$ to avoid excessive notation. The extension to $n > 2$ is straightforward.

First, $G(u)$ is componentwise concave iff

$$\begin{aligned} &G((u_1+\epsilon, u_2)) - G((u_1, u_2)) \\ &\quad \geqslant G\big((u_1+\epsilon+\alpha, u_2)\big) - G\big((u_1+\alpha, u_2)\big), \end{aligned} \tag{6}$$

for all nonnegative $u = (u_1, u_2)$, ϵ and α. By (2), and $E[Z_i(u_i)]$ being linear in u_i with the semigroup property, this is equivalent to

$$\begin{aligned} &E\big[V_0((Z_1(u_1+\epsilon), Z_2(u_2)))\big] - E\big[V_0((Z_1(u_1), Z_2(u_2)))\big] \\ &\quad \geqslant E\big[V_0((Z_1(u_1+\epsilon+\alpha), Z_2(u_2)))\big] \\ &\qquad - E\big[V_0((Z_1(u_1+\alpha), Z_2(u_2)))\big]. \end{aligned}$$

There exist three independent random variables Y_1, Y_2, and Y_3 that have the same distribution as $Z_1(u_1)$, $Z_1(\epsilon)$, and $Z_1(\alpha)$, respectively. Now, since $Z_1(\cdot)$ has the semigroup property, $Z_1(u_1+\epsilon)$ has the same distribution as $Y_1 + Y_2$ and $Z_1(u_1+\epsilon+\alpha)$ has the same distribution as $Y_1+Y_2+Y_3$.

Thus, we need to show

$$\begin{aligned} &E\big[V_0((Y_1+Y_2, Z_2(u_2))) - V_0((Y_1, Z_2(u_2)))\big] \\ &\quad \geqslant E\big[V_0((Y_1+Y_2+Y_3, Z_2(u_2))) - V_0((Y_1+Y_3, Z_2(u_2)))\big]. \end{aligned}$$

But this is clearly true because for any realization $Y_1 = k_1$, $Y_2 = k_2$, $Y_3 = k_3$ and $Z_2(u_2) = l$ we have by the concavity of $V_0(z)$ that

$$\begin{aligned} &V_0((k_1+k_2, l)) - V_0((k_1, l)) \\ &\quad \geqslant V_0((k_1+k_2+k_3, l)) - V_0((k_1+k_3, l)). \end{aligned}$$

Taking expectations on both sides with respect to all the random variables Y_1, Y_2, Y_3 and $Z_2(u_2)$, preserves this inequality, so (6) holds.

We next show $G(u)$ is submodular in a similar way. It is enough to show

$$\begin{aligned} &G\big((u_1+\epsilon_1, u_2+\epsilon_2)\big) - G\big((u_1, u_2+\epsilon_2)\big) \\ &\quad \leqslant G\big((u_1+\epsilon_1, u_2)\big) - G((u_1, u_2)), \end{aligned} \tag{7}$$

or

$$\begin{aligned} &E\big[V_0((Z_1(u_1+\epsilon_1), Z_2(u_2+\epsilon_2)))\big] \\ &\qquad - E\big[V_0((Z_1(u_1), Z_2(u_2+\epsilon_2)))\big] \\ &\quad \leqslant E\big[V_0((Z_1(u_1+\epsilon_1), Z_2(u_2)))\big] \\ &\qquad - E\big[V_0((Z_1(u_1), Z_2(u_2)))\big] \end{aligned} \tag{8}$$

for any nonnegative $u = (u_1, u_2)$, ϵ_1 and ϵ_2.

Let Y_{11}, Y_{12}, Y_{21}, and Y_{22} be independent random variables with the same distribution as $Z_1(u_1)$, $Z_1(\epsilon_1)$, $Z_2(u_2)$, and $Z_2(\epsilon_2)$, respectively. Using the semigroup property,

$$\begin{aligned} &E\big[V_0((Z_1(u_1+\epsilon_1), Z_2(u_2)))\big] \\ &\quad = E\big[V_0((Y_{11}+Y_{12}, Y_{21}))\big]. \end{aligned}$$

Similarly, we have

$$E\big[V_0((Z_1(u_1), Z_2(u_2+\epsilon_2)))\big] = E\big[V_0((Y_{11}, Y_{21}+Y_{22}))\big],$$

and

$$\begin{aligned} &E\big[V_0((Z_1(u_1+\epsilon_1), Z_2(u_2+\epsilon_2)))\big] \\ &\quad = E\big[V_0((Y_{11}+Y_{12}, Y_{21}+Y_{22}))\big]. \end{aligned}$$

Then, we can write the inequality in (8) as

$$\begin{aligned} &E\big[V_0((Y_{11}+Y_{12}, Y_{21}+Y_{22})) - V_0((Y_{11}, Y_{21}+Y_{22}))\big] \\ &\quad \leqslant E\big[V_0((Y_{11}+Y_{12}, Y_{21})) - V_0((Y_{11}, Y_{21}))\big]. \end{aligned}$$

Again, we only to need to make a sample path comparison to see that the above inequality holds. Take any realization z_1, k_1, z_2, and k_2 of the random variables Y_{11}, Y_{12}, Y_{21}, and Y_{22}, respectively, and we know

$$\begin{aligned} &V_0\big((z_1+k_1, z_2+k_2)\big) - V_0\big((z_1, z_2+k_2)\big) \\ &\quad \leqslant V_0\big((z_1+k+1, z_2)\big) - V_0\big((z_1, z_2)\big) \end{aligned}$$

holds because V_0 is submodular. Therefore, (7) holds and $G(u)$ is submodular with respect to u_1 and u_2. □

The above structural results give useful insights into the optimal joint-booking policies. The componentwise concavity of the expected net revenue function implies that there are critical booking levels for each reservation class beyond which the expected value does not increase, provided booking levels of other reservation classes are kept constant. The submodularity property implies that these optimal booking levels for each reservation class are nonincreasing in the level of bookings accepted for any other class. These are natural and intuitive properties. They simply reflect the fact that low reservation levels in one class mean that capacity will be less constrained in the service period, and this in turn reduces the potential costs of overbooking in other classes because more (or at least less costly) substitution options will be available.

4. Optimization

In this section, we provide procedures to compute the optimal overbooking levels. The problem in (1) has simple constraints (only lower bounds on the decision variables) and a two-stage objective function. The model resembles two-stage stochastic programming problems, for which there are well-known solution procedures (see Birge and Louveaux 1997). However, the recourse function is complicated in our case: The probability distribution is a nonlinear function of the decision variables. In that respect, we propose stochastic gradient algorithms to solve the problem and discuss their properties in this section. The submodularity property discussed above and the algorithm are used to define heuristic solution methods later in §5.

4.1. Poisson Approximation

The binomial model is a good model of real cancellation processes, but it is not well suited to continuous optimization methods. Therefore, in this section we consider a Poisson model that has continuous on-hand reservation values u_i. Specifically, we assume $Z_i(u_i)$ is a Poisson random variable with mean $p_i u_i$.

It is well known that there are certain problems with using this Poisson model of survivals. In particular, the number of surviving customers, $Z_i(u_i)$, could exceed the booking level; equivalently, the number of cancellations, $u_i - Z_i(u_i)$, can take on negative values. Such outcomes are clearly unrealistic. However, there are well-established bounds on the Poisson approximation to the binomial distribution (see Ross 1996), and it becomes arbitrarily good as u_i becomes large. Therefore, for problems with large booking levels, the approximation is justified. Also, the binomial model can be used heuristically in essentially the same continuous optimization method as described below.

4.2. Stochastic Gradient Algorithm

Our optimization algorithm is based on using an estimator of the gradient of the objective function $G(u)$ (a *stochastic gradient*) within a gradient projection iteration. Let a vector D^k denote the estimator of the gradient at the kth iteration of the algorithm. The algorithm requires a sequence of step sizes, $\{b_k\}$, satisfying

$$\sum_{k=1}^{\infty} b_k = +\infty, \quad \sum_{k=1}^{\infty} b_k^2 < +\infty;$$

for example, $b_k = 1/k$. Then, the algorithm proceeds as follows:

Step 0. Initialize: $k = 1$ and $u^k := x$.

Step 1. Get the stochastic gradient:

- Randomly generate a new vector $Z(u^k)$.
- Compute the gradient estimate D^k.

Step 2. Compute $u^{k+1} = \Pi(u^k + b_k D^k)$, where $\Pi(\cdot)$ is the projection of $u^k + b_k D^k$ onto $\{u\colon u \geqslant x\}$.

Step 3. Set $k := k+1$ and GOTO Step 1.

In Step 2 of the algorithm, one can even define a very large (redundant) upper bound vector $\bar{U} < \infty$ such that $\Pi(\cdot)$ is the projection onto $\{u\colon \bar{U} \geqslant u \geqslant x\}$. The upper bound can be chosen such that the optimal overbooking level is not excluded from the constraint set and the upper bound is greater than any reasonable upper bound on the demand. Further discussion on this topic is provided in §5. In the next two sections, we show how to compute gradient estimates D^k. We then provide convergence guarantees on the algorithm.

4.3. Gradient Estimates

To compute D^k, we need to estimate the following partial derivatives:

$$\frac{\partial}{\partial u_i} E[V_0(Z(u))] = \lim_{h\to 0} \frac{1}{h} [E[V_0(Z(u + e_i h))] - E[V_0(Z(u))]]. \tag{9}$$

We use the notation e_i for the ith unit vector in R^n. Let $Y_i(h)$ denote a Poisson random variable with mean $p_i h$ that is independent of $Z(u)$. Then, it is straightforward to show that

$$\begin{aligned} E[V_0(Z(u+e_i h))] &= E[V_0(Z(u) + e_i Y_i(h))] \\ &= E[V_0(Z(u)) \mid Y_i(h) = 0] P(Y_i(h) = 0) \\ &\quad + E[V_0(Z(u) + e_i) \mid Y_i(h) = 1] \\ &\quad \cdot P(Y_i(h) = 1) + o(h). \end{aligned}$$

Because $P(Y_i(h) = 1) = p_i h + o(h)$ and $P(Y_i(h) = 0) = 1 - p_i h + o(h)$, we obtain

$$\begin{aligned} &\frac{\partial}{\partial u_i} E[V_0(Z(u))] \\ &\quad = \lim_{h\to 0} \frac{1}{h} [p_i h [E[V_0(Z(u) + e_i)] - E[V_0(Z(u))]] + o(h)] \\ &\quad = p_i E[V_0(Z(u) + e_i) - V_0(Z(u))]. \end{aligned}$$

Therefore, an unbiased estimate of the partial derivative is

$$H_i(Z(u)) = p_i (V_0(Z(u) + e_i) - V_0(Z(u))).$$

Letting $H(Z(u)) = (H_1(Z(u)), \ldots, H_n(Z(u)))$, we have the following unbiased estimator of the gradient of $E[V_0(Z(u))]$:

$$\nabla_u E[V_0(Z(u))] = E[H(Z(u))].$$

The estimator $H(Z(u))$ can be computed by simulating $Z(u)$ and solving a network linear program to obtain $V_0(Z(u))$. Then, each estimate $V_0(Z(u) + e_i)$, $i = 1, \ldots, n$, can be computed by increasing $Z_i(u)$ by one and resolving the network problem. Let z^k be the realization of the vector $Z(u^k)$ at the kth iteration of the algorithm. Therefore, the components of the gradient estimate at iteration k are

$$D_i^k = r_i - q_i(1 - p_i) + p_i (V_0(z_i^k + e_i) - V_0(z_i^k)), \quad i = 1, \ldots, n. \tag{10}$$

We call the estimate (10) the "Difference Gradient" (DG).

Notice that by taking expectations on the right-hand side of (2) and using $E[Z_i(u_i)] = p_i u_i$, we can write the objective function in open form as

$$G(u) = \sum_{i=1}^n r_i(u_i - x_i) - \sum_{i=1}^n q_i (1 - p_i) u_i + \sum_{z_1=0}^{\infty} \cdots \sum_{z_n=0}^{\infty} V_0(z) \cdot P(Z_1(u_1) = z_1) \cdots P(Z_n(u_n) = z_n). \tag{11}$$

DG can then be obtained algebraically from (11), as well (see the appendix for a small example).

Another estimate of the derivative (9) can be obtained by the score function (or likelihood ratio) method (see Rubinstein and Shapiro 1993). Note that

$$\frac{\partial}{\partial u_i} P(Z_i(u_i) = z_i) = \left(\frac{z_i}{u_i} - p_i \right) P(Z_i(u_i) = z_i).$$

By differentiating the last term in (11) and substituting in the above expression, we obtain

$$\frac{\partial}{\partial u_i}E[V_0(Z(u))]=E\left[\left(\frac{Z_i(u_i)}{u_i}-p_i\right)V_0(Z(u))\right].$$

Therefore, an unbiased estimate of the derivative is

$$H_i(Z(u))=\left(\frac{Z_i(u_i)}{u_i}-p_i\right)V_0(Z(u)).$$

Therefore, the gradient at the kth iteration is

$$D_i^k=r_i-q_i(1-p_i)+\left(\frac{z_i^k}{u_i^k}-p_i\right)V_0(z^k),\quad i=1,\ldots,n. \quad (12)$$

We call the estimate (12) the "Score Function Gradient" (SFG).

Note that SFG requires solving the transportation problem only once to get D_i for $i=1,\ldots,n$, whereas DG requires solving it $(n+1)$ times. Therefore, computing one gradient estimate based on SFG is faster than computing one based on DG. However, because the variances of these estimators may differ, it is not clear if SFG results in a faster overall algorithm. Unfortunately, we have no theory to guide us here. However, in §5 we present some computational tests that compare the performance of SFG and DG. Our tests and experience indicate that the algorithm based on DG is in fact much faster, despite the fact that DG is more costly to compute at each iteration. (The reason appears to be that SFG has a much higher variance.)

We will also need the following continuity result regarding the objective function $G(u)$ (see the appendix for the proof).

Lemma 3. *If for each $i=1,\ldots,n$, $Z_i(u_i)$ is a Poisson-distributed random variable with mean p_iu_i, then the objective function, $G(u)$, is twice continuously differentiable.*

The next lemma establishes that both the DG and SFG estimates have finite variances (see appendix for the proof).

Lemma 4. *Given u^k at each iteration of the algorithm, there exists a finite constant C such that* $\mathrm{Var}(D_i^k)<C<\infty$ *for $i=1,2,\ldots,n$ for both DG and SFG.*

4.4. Convergence with DG Estimator

We next show that the algorithm of §4.2 with the DG gradient estimator has fairly robust local convergence properties. (We have not been able to verify whether Condition A3 below holds for the SFG estimator, so we cannot give convergence guarantees for this version of the algorithm.) Kushner and Clark (1978) provide a proof of convergence in probability for stochastic gradient projection algorithms. We show that our problem and algorithm with the DG estimator satisfy the properties described by Kushner and Clark.

The gradient D^k in our algorithm is a random vector and is in fact a "noisy" representation of the gradient of the function $G(\cdot)$. Let the noise (error) in the gradient at iteration k be the vector ξ^k. We have $\xi_i^k=D_i^k-(\partial/\partial u_i)G(u^k)$ for $i=1,\ldots,n$. Then, $E[\xi^k\mid u^1,\ldots,u^k]=0$ with probability one.

Let the cumulative step sizes be defined as $t_k=\sum_{i=1}^{k-1}b_i$ and define a function $m(t)$ such that $m(t)=\max\{k:t_k\leqslant t\}$ for $t\geqslant 0$, and $m(t)=0$ otherwise. Suppose the following assumptions hold:

A1. *$\{b_k\}$ is a sequence of positive real numbers such that $b_k>0$, $b_k\to 0$, and $\sum b_k=\infty$.*

A2. *Let the constraint set for the problem be defined by $\Theta=\{u\colon \theta_j(u)\leqslant 0\ j=1,\ldots,s\}$. The set Θ is closed and bounded. The $\theta_j(\cdot)$, $j=1,\ldots,s$, are continuously differentiable. At each u that is on the boundary of Θ, the gradients of the active constraints are linearly independent.*

A3. $\lim_{k\to\infty}P[\sup_{m(t_k+t)\geqslant l\geqslant k}|\sum_{i=k}^{l}b_i\xi^i|>\epsilon]=0$ *for each $\epsilon>0$ and $t>0$.*

A4. *$G(\cdot)$ is a continuously differentiable real-valued function on $\Re^n$.*

A5. *$b_kE|\xi^k|^2\to 0$ as $k\to\infty$.*

Let KT be the set of Kuhn-Tucker points of the problem in (1). (Note that the Kuhn-Tucker conditions are necessary for our problem (1) because the constraint qualifications always hold: The gradient of the left-hand side of the constraints, $u\geqslant x$, always has rank n.) Then, Theorem 6.3.1 of Kushner and Clark (1978) shows that if KT is a connected set and Assumptions A1–A5 hold, $\{u^k\}\to KT$ in probability as $k\to\infty$.

A weaker (and somewhat more technical to state) convergence result holds when KT is not connected. In order to do that, we first define an interpolation for the sequence $\{u^k\}$. We define the continuous function $u(t)$ by

$$u(t)=\begin{cases}u^k, & t=t_k,\\ \dfrac{t_{k+1}-t}{b_k}u^k+\dfrac{t-t_k}{b_k}u^{k+1}, & t\in(t_k,t_{k+1}).\end{cases}$$

(Note that $u(t)$ is just a linear interpolation of the values u^k as a function of the cumulative step sizes t_k.) Let $N_\epsilon(KT)$ denote the epsilon neighborhood of the set KT and let $I(x,S)$ be the indicator of $x\in S$ (i.e., $I(x,S)=1$ if $x\in S$, and $I(x,S)=0$ otherwise). Kushner and Clark (1978), Theorem 6.3.1 shows that under A1–A5, if KT is not connected, then for each $\delta>0$ and $\epsilon>0$ there exists a $t_0<\infty$ such that $t>t_0$ implies

$$\lim_{k\to\infty}P\left(\frac{1}{2t}\int_{-t}^{t}I(u(t_k+s),\Re^n-N_\epsilon(KT))\,ds\geqslant\delta\right)\leqslant\delta.$$

If KT is bounded, the last δ on the right-hand side above can be replaced by zero. Roughly, this result says that the "average amount of time" the iterates u_k lie more than ϵ away from a point in KT (averaging over a sufficiently large but finite interval) becomes arbitrarily small as k

increases. It is, in essence, a convergence in probability of a "moving average" of u^k's rather than a convergence of u^k itself. (Indeed, we use such a moving average of iterates in our implementation of the algorithm in §5.)

Next, we discuss that Assumptions A1–A5 hold for our algorithm with the DG estimator. A1 is satisfied by our choice of the step sizes b_k in §4.2. A2 is satisfied because the constraint set in our problem is modified to take the form $\{u\colon \bar{U} \geqslant u \geqslant x\}$. A3 holds for the DG estimator by choice of b_k; boundedness of D^k, ξ^k, $E[D^k]$, $\nabla_u G(\cdot)$; and by Lemma 4. A4 holds by Lemma 3. A5 holds by choice of b_k and boundedness of $\mathrm{Var}(D^k)$ for the DG estimator and can be shown similar to the proof of Lemma 4. For SFG, A3 may not hold because, D^k is not bounded.

As a summary, the algorithm we proposed with the DG estimator satisfies the requirements for the convergence to a *KT* point. When *KT* is connected, we have convergence in probability. Even if *KT* is not connected, we still have a guarantee of convergence of the average of iterates to a point arbitrarily close to *KT*. Again, however, we emphasize that they are local convergence guarantees only.

4.5. Heuristic Extensions of the Algorithm

In this section, we present variations of the stochastic gradient algorithm that apply to (i) binomial distributed cancellations and (ii) adjustments of booking levels by simultaneously looking at random demand.

4.5.1. A Heuristic Gradient Algorithm for the Binomial Model. As discussed above, assuming survivals are Poisson distributed has some disadvantages. In particular, the number of survivors can be more than the number of reservations. This can lead to undesirable results, such as obtaining overbooking levels that are strictly less than capacity to "hedge" against the possibility of survivals exceeding capacity.

If we use the binomial model, so that $Z_i(u_i)$ is binomial distributed with parameters p_i and u_i, then the objective function can be written in the open form as follows:

$$G^{\text{bin}}(u) = \sum_{i=1}^{n} r_i(u_i - x_i) - \sum_{i=1}^{n} q_i(1-p_i)u_i + \sum_{z_1=0}^{u_1} \cdots \sum_{z_n=0}^{u_n} V_0(z) \cdot P(Z_1(u_1)=z_1)\cdots P(Z_n(u_n)=z_n), \tag{13}$$

where

$$P(Z_i(u_i)=z_i) = \frac{u_i!}{z_i!(u_i-z_i)!} p_i^{z_i}(1-p_i)^{(u_i-z_i)}$$

for $i = 1, \ldots, n$. In this case, the overbooking levels should be treated as discrete variables, which of course introduces other complications into the optimization.

However, consider the difference function for the function $G(u)$ under the binomial model. We next show:

Lemma 5. *If $Z_i(u_i)$ is binomial distributed with parameters p_i and u_i for $i = 1, \ldots, n$, then*

$$\begin{aligned}\Delta_i G^{\text{bin}}(u) &= G^{\text{bin}}(u+e_i) - G^{\text{bin}}(u)\\ &= r_i - q_i(1-p_i)\\ &\quad + p_i E[(V_0(Z(u)+e_i) - V_0(Z(u)))].\end{aligned}$$

Proof. The first two terms are immediate from (13). Let Y_i be a Bernoulli-distributed random variable with parameter p_i. Then, $Z(u+e_i)$ has the same probability distribution as $Z(u)+e_iY_i$, given the semigroup property. Then, similar to the Poisson case in §4.3, the difference function with respect to u_i can be determined for the expected service period revenue as

$$\begin{aligned}\Delta_i E[V_0(Z(u))] &= E[V_0(Z(u+e_i))] - E[V_0(Z(u))]\\ &= E[V_0(Z(u)+e_iY_i) - V_0(Z(u))]\\ &= p_i E[V_0(Z(u)+e_i) - V_0(Z(u))].\end{aligned}$$

The same relation can be shown algebraically using the open form of the function $G^{\text{bin}}(\cdot)$ (see the appendix for a small example). □

Note that this expression for the difference function is identical to the DG gradient (10). Essentially, only the probability measure has changed. If we heuristically allow overbooking levels u to be continuous and treat the difference function as the "gradient" of G^{bin}, then the gradient estimate

$$D_i^k = r_i - q_i(1-p_i) + p_i(V_0(Z(u)+e_i) - V_0(Z(u))$$

can be used in the algorithm of §4.2 with binomial simulations (or mixtures of binomials for a good rounding procedure) replacing Poisson simulations as an optimization heuristic. This approach combines the advantages of continuous, gradient-based optimization while avoiding some of the bad behavior of the Poisson model. Section 5 provides some numerical examples illustrating the performance of this heuristic.

4.5.2. Booking Limits and Random Demand to Come. Because the stochastic gradient algorithm is simulation based, it is easily adaptable. We only mention one basic variation here, which seems most important, but it should suggest others.

As mentioned above, the model and algorithm as stated do not account for limited demand to come. That is, it computes the overbooking *level* u, assuming that we will indeed be able to achieve u reservations on hand by the end of the reservation period. In reality, this may not be possible because of limited reservation demand.

One way to deal with this in the algorithm is to simultaneously simulate both the number of reservation requests and the number of cancellations. The modification proceeds as follows: First, redefine u_i to be an overbooking *limit*—rather than overbooking *level*. That is, u_i limits the number of requests for reservation class i that we are willing to hold to u_i if demand exceeds u_i. Let Y_i, a random variable, represent the demand to come for reservations in class i, and recall x_i is the number of reservations on hand at the start of the reservation period. Then, the number of reservations on hand at the end of the reservation period will be $\min\{u_i, Y_i + x_i\}$—that is, the minimum of the overbooking limit u_i and the number of reservations we have on hand plus the number of new reservations we receive. If $Y_i + x_i > u_i$, then a small increase in u_i will result in more reservations on hand—and potentially more surviving customers. However, if $Y_i + x_i \leqslant u_i$, an increase in u_i will not result in any more reservations on hand—and no increase in the number of surviving customers. Let D_i^k be any of the gradient estimates mentioned above. Then, one can modify D_i^k to account for random demand to come by simulating a value Y_i and using the modified estimator

$$\hat{D}_i^k = \begin{cases} D_i^k, & Y_i > u_i - x_i, \\ 0, & Y_i \leqslant u_i - x_i. \end{cases}$$

That is, the gradient estimate is simply set to zero in cases where the simulated value $Y_i \leqslant u_i - x_i$, which reflects the fact that an increase in the overbooking limit for such realizations does not result in any change in net revenues.

5. Numerical Examples

In this section, we report on some numerical tests of the stochastic gradient algorithm. We first solve a series of examples in which we test the speed of the algorithm using both the DG and SFG gradient estimates. Our tests show that DG is considerably more efficient.

Because the DG proved to be much more efficient, we used only DG in our remaining tests. These tests involve solving several "dynamic" examples, in which overbooking levels are computed periodically throughout a simulated booking process. This type of test mimics how static overbooking models are used in practice. We test several variations of the algorithm against simpler heuristics. The examples serve several purposes. First, they illustrate the potential performance improvement from using a substitution-based method rather than more naive heuristics. Second, they provide some guidance about the possible implementation choices in our algorithm. Finally, they show the effect of substitutability on net revenues, overbooking levels, and service measures.

5.1. Step-Size Rule

In implementing the stochastic gradient algorithm of §4, we used an online step-size rule suggested by Ruszczynski and Syski (1986). The step sizes are computed recursively by

$$b_0 > 0, \quad b_k = b_{k-1} e^{-\alpha\gamma_k - \delta b_{k-1}}, \quad k = 1, 2, \ldots,$$

where

$$\gamma_k = (D^k)^T \Delta u^k + \lambda (\Delta u^k)^T (\Delta u^k),$$

for $\Delta u^k = u^k - u^{k-1}$ and $\alpha > 0$, $\lambda > 0$ fixed. We used parameter values of $\alpha = 1$, $\delta = 2$, and $\lambda = 1$. This rule is motivated by using the information gathered during the course of the algorithm to update the step sizes and to enhance the speed of convergence far from the solution. It is proven that the step sizes, b_k, determined by this online rule behave asymptotically as the deterministic step-size rule $1/\delta(k+1)$ when the objective function is twice continuously differentiable. In Lemma 3, we proved that the function G is continuously differentiable. We stopped the algorithm when we reached a prespecified number of iterations or a prespecified minimum step size. Our stochastic gradient algorithm and a primal-dual based solution procedure to solve the transportation problem are both coded in C and run on a Pentium-III machine under the WindowsNT operating system.

5.2. The Computational Speed of DG and SFG

To test the relative speed of the algorithm under the two gradient estimates, we used a series of examples based on overbooking back-to-back flights. This problem arises when there are multiple departures on the same route. Customers on oversold flights can then be put on later-departing flights. This is only a one-way substitution, because one flight is a substitute for another only if its departure time is later. Such substitutions usually require compensation for the passenger and some loss of goodwill.

Our experiment involves $n = 2, 4, 10, 20$ flights, which allows us to test the performance of DG and SFG on problems of varying size. The parameters are the same for each of the flights, except for the substitution costs. The unit revenue is \$500, the unit overbooking cost is \$1,000, and the survival probability for the reservations is 0.9. We assume it is possible to substitute only one, two, or three flights ahead (i.e., a passenger for a flight can be delayed and served only by the next three flights on the airline's schedule). The substitution costs are \$600 to the next flight, \$700 to the second flight, and \$800 to the third flight.

We ran the two versions of the algorithm for 100,000 iterations, recording the average overbooking levels and total computation time every 2,000 iterations to track their progress. While one can use several criteria for evaluating convergence to an optimal set of booking levels u^*, we chose the distance metric

$$\|u - u^*\| = \sqrt{\sum_{i=1}^{n} (u_i - u_i^*)^2}.$$

In computing this metric, we rounded the iterates at each stage (e.g., after every 2,000 iterations) and used the optimal integer booking levels u^*. We did this because the algorithm ultimately is used for finding integer booking levels,

Table 1. Average booking levels and computation time (sec.) for SFG and DG.

Number of	Booking Levels		Time (sec)	
Iterations	SFG	DG	SFG	DG
2,000	(571, 683, 502, 499)	(121, 119, 118, 115)	4	5
4,000	(247, 726, 379, 289)	(120, 118, 117, 115)	8	9
6,000	(136, 648, 212, 200)	(120, 117, 117, 115)	12	13
8,000	(168, 472, 149, 131)	(120, 118, 117, 115)	16	18
10,000	(122, 432, 94, 94)	(120, 118, 117, 115)	19	22
20,000	(119, 281, 108, 106)	(120, 118, 117, 115)	39	44
30,000	(121, 198, 83, 106)	(120, 118, 117, 115)	59	65
40,000	(118, 150, 95, 116)	(120, 118, 117, 115)	79	86
50,000	(119, 124, 113, 114)	(120, 118, 117, 115)	99	108
60,000	(120, 118, 114, 114)	(120, 118, 117, 115)	119	130
70,000	(119, 118, 116, 114)	(120, 118, 117, 115)	139	151
80,000	(120, 116, 116, 115)	(120, 118, 117, 115)	159	173
90,000	(120, 118, 117, 115)	(120, 118, 117, 115)	179	194
100,000	(119, 119, 117, 115)	(120, 118, 117, 115)	198	216

so the integer solutions produced are what matter most. The optimal integer solution u^* was identified by simulating several integer solutions near the convergent point and picking the one with the highest net revenues. In general, several neighboring booking levels often gave very similar revenues, so it is possible that a deviation from u^* does not produce a significant loss in net revenue. Moreover, it is possible that our procedure misses the optimal integer booking levels, because we did not exhaustively test all possible integer values.

One can improve the convergence of the algorithms by adding upper bounds (denoted $\overline{U}_i$) on the overbooking levels at each iteration. This guarantees the overbooking levels generated by the stochastic gradient algorithm are between x_i and $\overline{U}_i$ at each iteration. If the dummy bound $\overline{U}_i$ is chosen to be small (but still large enough not to exclude the optimal overbooking level), the algorithms will tend to exhibit less variability and reach the optimal values faster. Our experiments showed that SFG has higher variance and its rate of convergence is more sensitive to choice of $\overline{U}_i$. After some initial experiments, we used $\overline{U}_i = 1{,}000$ for all i, unless noted otherwise.

To illustrate our findings, the results for the problem with four flights are summarized in Table 1. The SFG estimator required 50,000 iterations and 99 seconds of CPU time to find acceptable booking levels. In contrast, the DG estimator required only 2,000–4,000 iterations and less than 9 seconds to find acceptable booking levels.

The CPU time as a function of the number of iterations of the stochastic gradient algorithm is graphed in Figure 1 for all cases tested. As discussed earlier, DG solves $n+1$ network problems to compute a gradient estimate at each

Figure 1. CPU time as a function of the number of iterations.

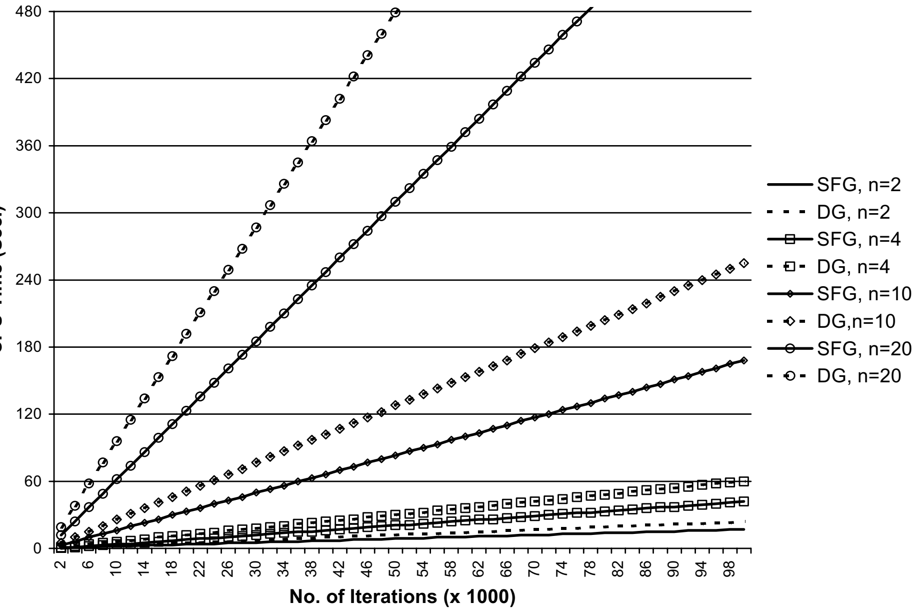
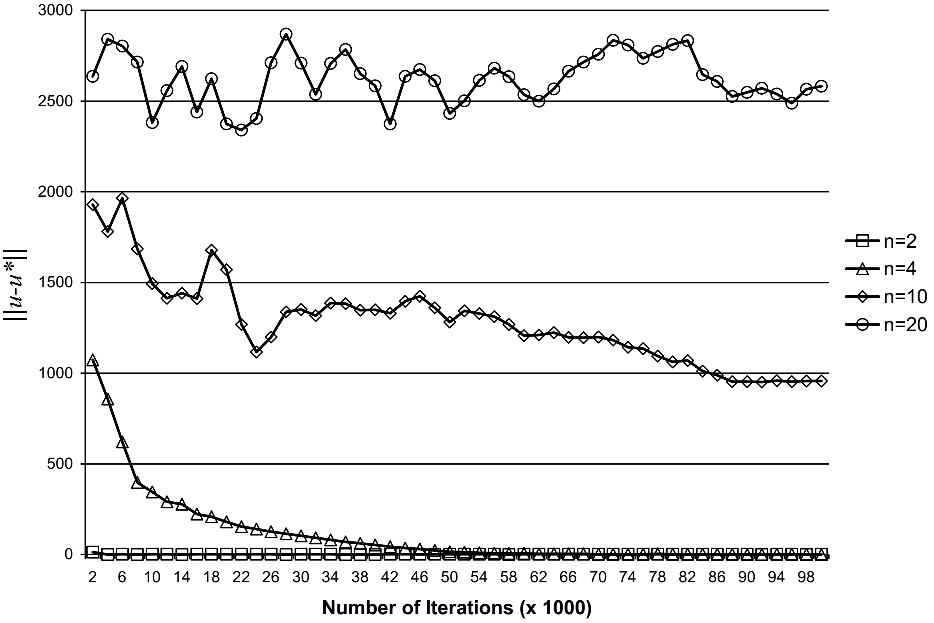


Figure 2. Distance to optimum u^* as a function of the number of iterations: SFG.



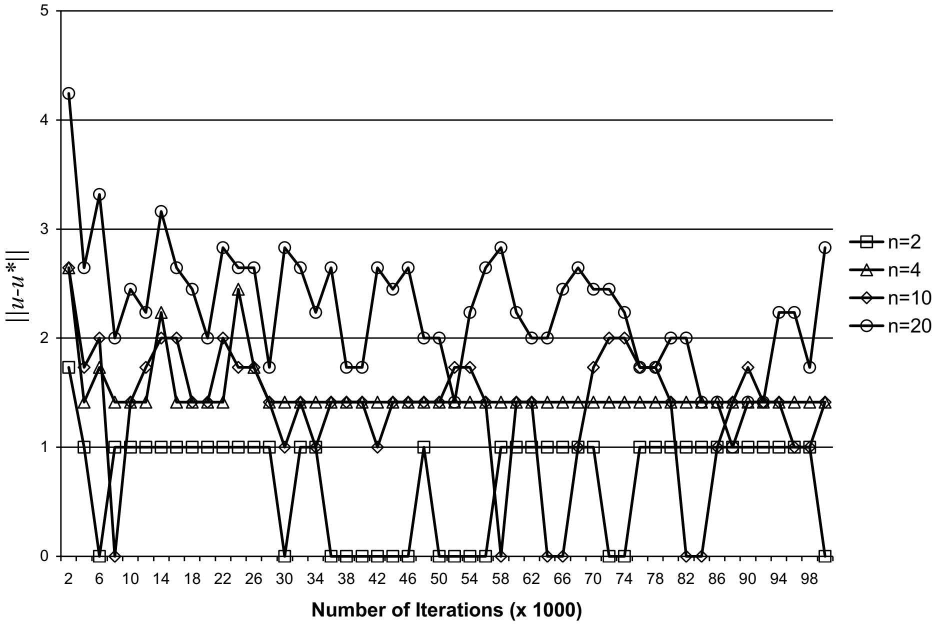
iteration of the algorithm, whereas SFG solves only one. This is reflected in the overall computation time, DG being slower, and the difference in computation times between two methods increasing as n increases.

The distance of the obtained overbooking levels to optimal integer solution are plotted in Figure 2 and Figure 3 for SFG and DG, respectively. As a result of our experiment in Example 1, DG seems to give more robust and reliable results, faster, than SFG. This is no surprise, because we have not been able to show that SFG indeed satisfies the theoretical requirements for the convergence of the stochastic gradient algorithm. In particular, SFG performs poorly as the number of inventory classes increases in our example. For these reasons, we abandoned the SFG estimator and used only the DG estimator in the remainder of our experiments.

5.3. Implementation Variations of the Stochastic Gradient Method

We tested several implementations of the stochastic gradient method. All variations used the DG estimator. The variations are:

Figure 3. Distance to optimum u^* as a function of the number of iterations: DG.



SOPT—This version uses the Poisson model of cancellations with continuous booking levels as described in §4.2.

BIN—This policy is the same as SOPT except that it uses binomial-distributed survivals, rather than Poisson. It uses the DG gradient estimate (see the discussion in §4.5.1) and the same step-size rule to update the booking level at each iteration. The overbooking levels are rounded to nearest integer values at each iteration.

SOPTUB—Due to the submodularity of the expected net revenue function, if bookings (demand) are (is) expected to be low in one class, then it makes sense to accept more reservations in the remaining classes. As described, SOPT does not allow for this sort of behavior. The policy SOPTUB attempts to capture this effect by adding roughcut upper bounds on the overbooking level in each class based on the expected demand to come. Specifically, we define an upper bound N_i for class i and change the constraints to $x_i \leqslant u_i \leqslant N_i$, $i = 1, \ldots, n$. After some experimentation, we set $N_i = x_i + \mu_i + 2.5\sigma_i$, where μ_i and σ_i are the mean and standard deviation of expected demand to come for class i. Thus, the likelihood that the number of arriving reservations for class i will exceed N_i is small. (Hence, class i bookings are effectively not constrained by this bound.) However, if the mean number of reservations for class i is small, then u_i will be limited and (by submodularity) the algorithm will tend to increase the booking levels of the classes $j \neq i$. These upper bounds serve to restrain the algorithm from "overallocating" capacity to a class for which we do not expect many reservations.

BINUB—Same as SOPTUB but uses the binomial distribution for the survivals.

5.4. Other Heuristic Policies

To test the absolute performance of these policies, we also simulated some simpler, naive heuristics. They are:

ADHOC—This policy computes independent overbooking for each class in the reservation period, but makes use of ad hoc upgrades at the time of service. That is, overbooking levels are set without accounting for the possibility of substitution, but when the service period arrives, substitution is indeed used to minimize the denied service and substitution costs. This roughly corresponds to the way substitution is managed in many real applications. Specifically, the ADHOC overbooking level for class i is chosen to maximize

$$G_i^A(u_i) = r_i u_i - q_i(1 - p_i)u_i + a_{i0} \sum_{z_i = c_i + 1}^{u_i} (z_i - c)P(Z_i(u_i) = z_i), \qquad (14)$$

where recall r_i is the unit revenue, q_i is the cancellation fee, p_i is the probability of survival, a_{i0} is the unit overbooking penalty, c_i is the capacity, and the random variable $Z_i(u_i)$ is the number of surviving reservations.

OBCOST—This policy is similar to ADHOC, but instead of using a_{i0} as the overbooking penalty, it uses an estimate of the "effective" cost of overbooking. Specifically, given u, and ad hoc use of upgrades, not every incidence of overselling results in denied boardings. It may result in substitution at a lower cost. Therefore, solving (14) to determine class i overbooking levels is too conservative when a_{i0} is used as the class i overbooking penalty. To correct for this, we use a cost $a(u)$, the effective overbooking cost being a function of the overbooking levels for all the classes. We compute this value by simulation; i.e., once we set u, we simulate the number of show-ups, solve for the service allocation problem, and compute the effective overbooking cost. Then, we use a search procedure to find the u that maximizes G^A.

POOLED—This policy combines the capacity from all classes and computes a single, aggregate overbooking level that is applied to the aggregate reservation level. This aggregate overbooking level is obtained by solving a problem identical to (14). The cancellation probabilities and cost parameters are chosen somewhat heuristically, as described in the examples below.

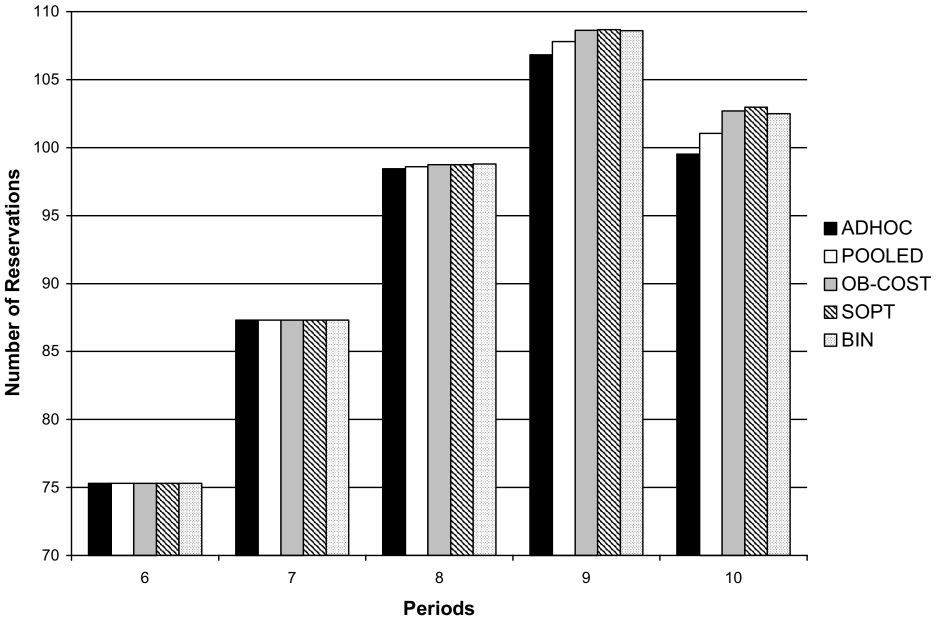
5.5. Performance in Some Simulated Applications

Example 1. In this example, we look at an overbooking problem for a single flight with two reservation classes (first and coach) and two physical inventory classes (first-class cabin and coach cabin). There are 20 seats for first class and 100 for the coach. A first-class ticket is sold for \$1,000, and there is no cancellation fee (i.e., refund on cancellation is \$1,000). Unit revenue for a coach-class reservation is \$200 and the cancellation fee is \$50 (i.e., refund is \$150). The unit overbooking cost for coach class is \$400. We assume there is no bonus for upgrading coach-class customers to first class. As in common airline practice, we do not allow for overbooking of first class (hence, no downgrading). This results in setting $u_1^* = 20$ initially. So, the problem reduces to finding the overbooking level for coach class only.

We assume the planning horizon consists of 10 periods, the last one being the service period. We assume the number of arriving reservations in each period is Poisson distributed and is homogeneous over the 10 periods. In each period, we compute the overbooking levels and observe the reservation requests that arrive in that period. We decide to accept or reject these requests based on the period's overbooking level. Requests are rejected if the total on-hand reservations plus the new requests are more than the overbooking level. Next, we observe the cancellations (reservations accepted in this period, and the ones remaining from the previous periods may be cancelled), and move on to the next period. In the last period (the service period), the surviving reservations are assigned to the inventory.

The cost and revenue parameters do not change over time, but the survival probabilities are increasing in the number of periods remaining. A first-class reservation made in any of the periods $1, 2, \ldots, 10$ survives to the service

Figure 4. Average surviving coach-class reservation levels for policies: Example 1.



period with a probability of $0.71, 0.72, \ldots, 0.80$, respectively. The corresponding survival probabilities for a coach-class reservation are $0.81, 0.82, \ldots, 0.90$ at the end of the periods $1, 2, \ldots, 10$, respectively. These values are chosen to be close to the ones cited in Smith et al. (1992). As mentioned earlier, we do not distinguish between cancellations and no-shows in our model. These probabilities are "survival-until-service" probabilities. (However, in this multiperiod simulation, it is easy to redefine the survival probabilities such that they represent the survivals from one period to the next. In that case, the cancellations between reservation periods are *cancellations*, whereas the cancellations at the end of the last reservation period are *no-shows*.) The "load factor" (the ratio of total expected demand net of cancellations to capacity) is chosen to be 1.10 for both classes. This corresponds to an average arrival rate of 30 requests for first class and 130 requests for coach over the entire planning horizon.

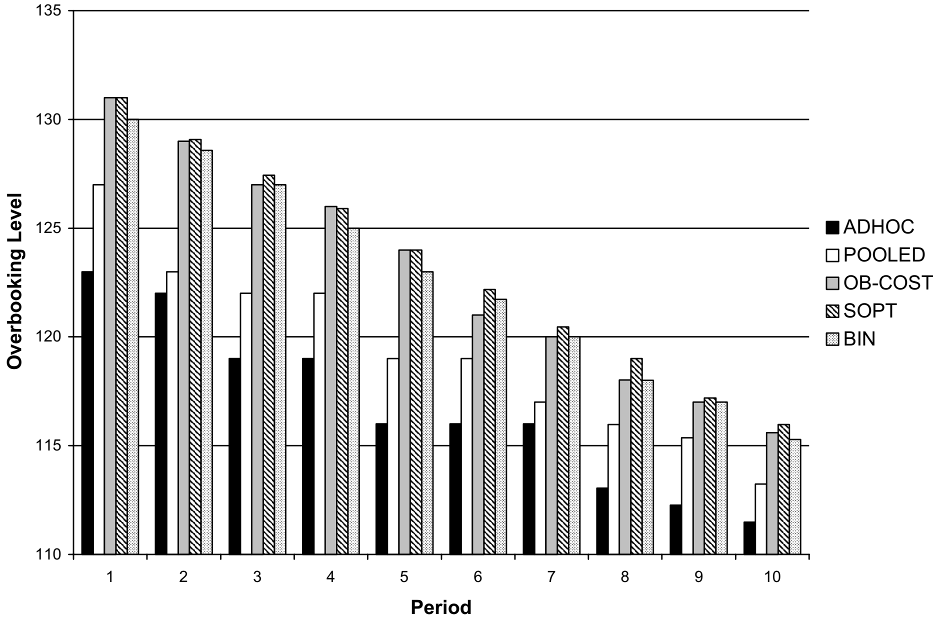
The policies ADHOC, POOLED, OBCOST, SOPT, and BIN were all evaluated using simulation. SOPTUB and BINUB were not used in this example. However, these policies are tested in the next example where demand for the first-class cabin is smaller. For the policy POOLED, the pooled class has the same parameters as the coach class, except that its survival probability is an average of first- and coach-class survival probabilities weighted by the first-class and coach cabin capacities. Given that the survival probabilities are lower than the ones used in ADHOC, the coach-class overbooking levels in each period are higher for POOLED, and this may lead to more upgrades and better use of the limited number of seats in the plane, though it might also lead to more denied boardings and increased cost.

We used common random numbers for the simulation of all policies and looked at their relative performance at the end of 500 independent trials. The average number of surviving reservations at the end of each reservation period are shown in Figure 4 (the number for the last period stands for the number of show-ups). The average overbooking levels per period are shown in Figure 5. The revenue and service performance of the policies are summarized in Table 2. We report the average revenues as well as 90% confidence intervals for the average revenue. We also provide the increase in revenues as a percentage of the average revenues of ADHOC policy. All the policies result in increase in average revenues, despite worse service performance compared to ADHOC. The service levels in Table 2 are computed only for the coach class; i.e., the %upgrade refers to the percentage of coach-class show-ups that were upgraded to first class, and the %denied is the percentage of coach-class customers that were denied service.

Note from Figure 5 that the overbooking levels are nonincreasing over time for each of the policies. This follows because the cancellation probabilities decrease over time, so we are willing to hold a higher number of reservations early on in the booking process. In the last period, SOPT has the highest overbooking levels, followed by OBCOST and BIN, all with more than 115 reservations. As a result, these policies have higher revenues than ADHOC and POOLED, despite the fact that %denied is higher for OBCOST, SOPT, and BIN. All the policies make use of upgrades for coach-class passengers, which vary from 1.15% in ADHOC to 2.41% in SOPT.

Example 2. This example is the same as Example 1 above, except that the demand for first class is lower. Specifically,

Figure 5. Average coach-class overbooking levels for policies: Example 1.



the demand for coach class is the same as in Example 1, but the expected demand for first class is only six booking requests over the entire planning horizon. Because the overbooking level of first class is fixed at the first cabin capacity in Example 1 and none of the policies ADHOC, POOLED, and OBCOST consider first and coach class jointly while setting the coach-class overbooking level, their overbooking levels are the same as in Example 1 and they accept the same number of coach reservations. However, there is a reduction in revenues due to lower first-class demand and reduction in overbooking costs due to more upgrades.

We used the policies SOPTUB and BINUB for this example. When demand for the first-class cabin is very low, SOPTUB and BINUB will allow more aggressive overbooking of coach reservations. As shown below, this is indeed the case.

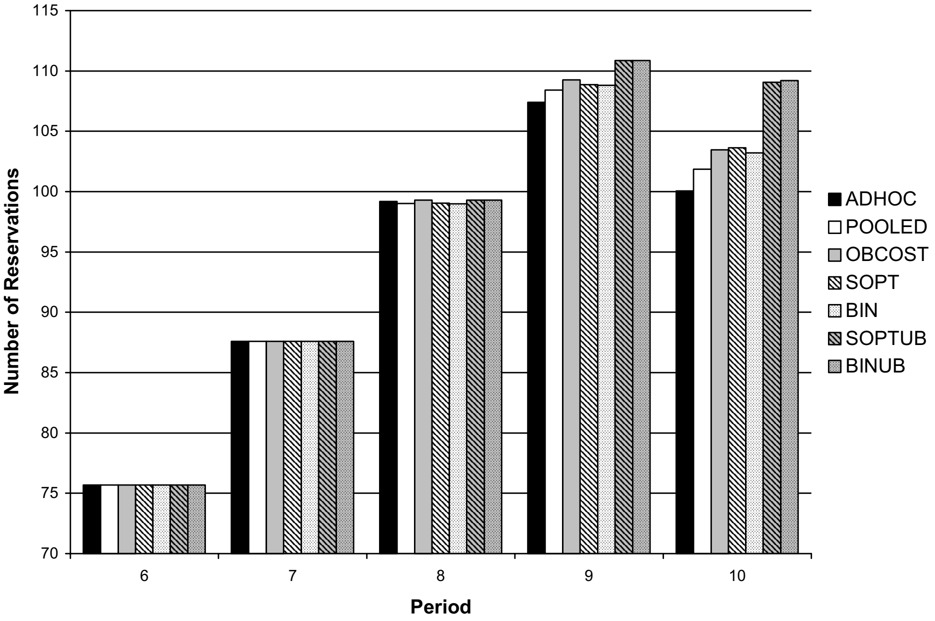
The average overbooking levels per period and average number of reservations in the system at the end of each period are given in Figures 6 and 7. The average revenues and service levels are given in Table 3. As expected, the highest revenues are observed for policies SOPTUB and BINUB. More than 8% of the coach-class passengers are upgraded in both of these policies, resulting in an increase in revenues. These are significant increases, with BINUB in particular achieving a net revenue about 6.8% higher than ADHOC. From Figures 6 and 7, note that BINUB and SOPTUB have the highest overbooking levels for coach class, allowing for more upgrades, which increase the average revenues significantly. The average number of survivors for first class is around 5.5 for all the policies. For coach class, the average number of survivors ranges from 100 for ADHOC; 102 for POOLED; 103 for OBCOST, SOPT, and BIN; to 109 for BINUB and SOPTUB. As a result, there are, on the average, 14 empty seats in the plane in ADHOC and only 6 empty seats in BINUB and SOPTUB.

Example 3. This example is a variant of the one used to test SFG and DG. An airline has four consecutive departures on the same route. Overbooking leads to substitution only forward in time; i.e., customers of an oversold flight can take later flights with some loss of goodwill. We assume that denying service completely to a customer results in a higher cost than the cost of goodwill due to delays. For simplicity, we also assume that all four flights serve one booking class each, and each flight has the same capacity. Similarly, we denote the flights by 1, 2, 3, and 4,

Table 2. Average revenues and service levels for policies: Example 1.

	Revenue			Service Levels	
Policy	Avg.	90% Conf. Int.	%Over ADHOC	%Upgrade	%Denied
ADHOC	36,651	(36,515; 36,787)	0.0	1.15	0.22
POOLED	36,811	(36,681; 36,940)	0.44	1.59	0.59
OBCOST	36,868	(36,744; 36,992)	0.59	2.34	1.27
SOPT	36,868	(36,746; 36,990)	0.59	2.41	1.40
BIN	36,879	(36,756; 37,001)	0.62	2.20	1.14

Figure 6. Average number of surviving coach-class reservation levels for policies: Example 2.



ordered in time (i.e., the earliest flight is Flight 1). There are 100 seats available on each flight. Delaying service to customers by one flight has a cost of $300; delaying by two flights is $400; delaying by three flights is $500 (i.e., if a customer of Flight 1 is flown on Flight 4, the cost is $500, and if the customer of Flight 2 is flown on Flight 3, the cost is $300). The cost of denying service to a customer on any flight is $1,000. The unit revenue for reservations is $500, which is fully refundable upon cancellation. Similar to Examples 2 and 3, there are 10 reservation periods in the planning horizon.

The only naive heuristic we tried in this example was ADHOC. OBCOST was deemed too computationally difficult to implement. In Example 2, we were able to compute the effective overbooking cost for coach class at every iteration and period. Because the overbooking level for first class was fixed, we were doing a simple search on coach-class overbooking levels. However, in this example we have four different booking classes, and their overbooking levels are neither fixed nor known prior to simulation. This makes it difficult to compute an accurate effective overbooking cost. As for POOLED, because the problem parameters are

Figure 7. Average coach-class overbooking levels for policies: Example 2.

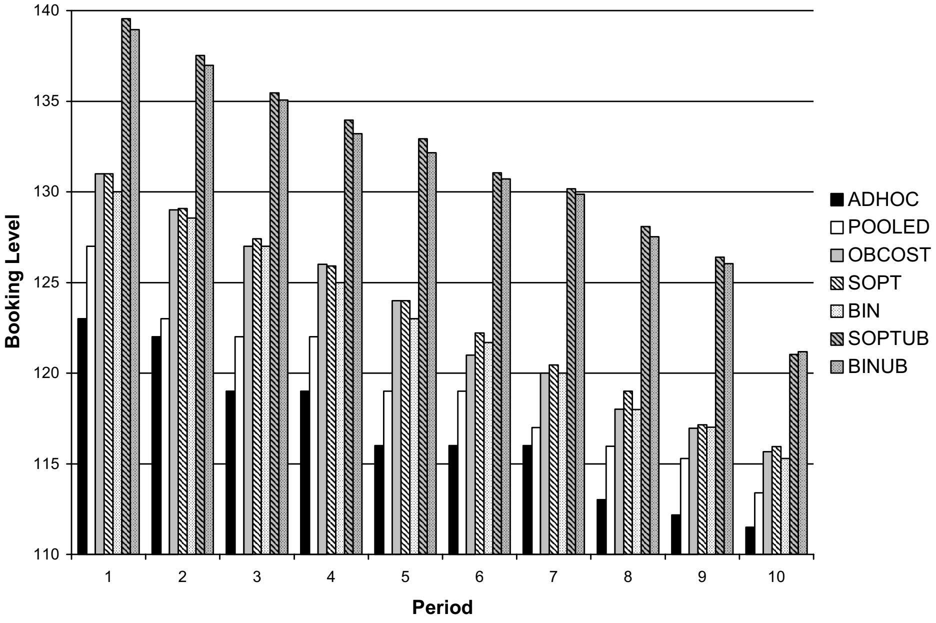


Table 3. Average revenues and service levels for Example 2.

	Revenue			Service Levels	
Policy	Avg.	90% Conf. Int.	%Over ADHOC	%Upgrade	%Denied
ADHOC	25,452	(25,113; 25,790)	0.0	1.28	0.0
POOLED	25,816	(25,469; 26,163)	1.43	2.34	0.0
OBCOST	26,140	(25,791; 26,489)	2.70	3.78	0.0
SOPT	26,202	(25,855; 26,548)	2.95	4.06	0.0
BIN	26,115	(25,852; 26,378)	2.60	3.66	0.0
SOPTUB	27,124	(26,766; 27,481)	6.57	8.50	0.32
BINUB	27,189	(26,839; 27,538)	6.82	8.52	0.31

symmetric across the flights, this is no different than an independent overbooking policy similar to ADHOC. We did try all variations of the stochastic gradient algorithm, however—SOPT, BIN, SOPTUB, and BINUB.

We tried three scenarios. In Scenario 1, the survival probabilities are $0.81, 0.82, \ldots, 0.90$ from the first period to the last. The demand is assumed to be Poisson distributed in each period, and the load factor is 1.1. Each flight has the same demand distribution and demand is homogeneous over time. We used common random numbers for simulating the policies and report the results of 500 independent trials. The results for Scenario 1 are given in Table 4. For this example, only BIN provides revenues higher than ADHOC. The service levels reported for this example are the percentage delayed (substituted by later flights) and the percentage of denied service on all the flights. The average overbooking levels in the last period and average number of surviving customers (show-ups) are shown in Figure 8.

The overbooking levels for ADHOC for all the flights is around 110, and on average less than 100 passengers show up for the flights, with the %delayed and %denied boarding each being less than 0.75%. For stochastic gradient policies, the overbooking levels are decreasing over the flights, which is expected because of one-way substitution and submodularity. SOPT and SOPTUB suffer from high overbooking levels and higher number of show-ups; their service levels are worse and the net revenues are in fact somewhat less than the other policies. BIN and BINUB provide a slight advantage over ADHOC; they have higher overbooking levels compared to ADHOC, therefore slightly higher percentage substituted (delayed). But the differences are not large among any of the policies in this example.

In Scenario 2, we use a different demand pattern. We assume the first and the last flights are more popular than the second and third flights. The total expected demand for the system—net of cancellations—is the same as in Scenario 1, but the first and the last flights get 30% of the demand each, and the second and third flights get 20% each. The results are given in Table 5. As expected, SOPTUB and BINUB have the highest average revenues, as they use the asymmetric demand information. They have more than 1% increase in revenues compared to ADHOC, with more than 4% of passengers who show up being served by later flights. The average overbooking levels in the last period and average number of surviving customers (show-ups) are shown in Figure 9 for all the policies.

Scenario 3 is a quite extreme example of asymmetric demand. This time, the average number of requests per period is 30, 3, 3, and 3 for Flights, 1, 2, 3, and 4, respectively. This extreme asymmetry highlights the effect of substitution and the use of upper bounds in determining the overbooking levels. The results are given in Table 6, and the average overbooking levels in the last period and average number of surviving customers (show-ups) are shown in Figure 10.

Note from Table 6 that the number of denied boardings is negligible for all policies, but the %delayed reaches 46% for SOPTUB and BINUB. These policies take very large numbers of reservations on the first flight and then end up serving a large fraction of them on later flights (a sort of “bait-and-switch” policy). As a result, the net revenue of SOPTUB and BINUB is much higher than ADHOC—approximately 24% higher. Again, such extreme behavior is not realistic and in practice one would clearly limit it by

Table 4. Average revenues and service levels for different policies in Example 3, Scenario 1.

	Revenue			Service Levels	
Policy	Avg.	90% Conf. Int.	%Over ADHOC	%Delayed	%Denied
ADHOC	195,129	(194,944; 195,313)	0.0	0.41	0.74
SOPT	194,744	(194,445; 195,043)	−0.20	1.15	0.30
BIN	195,342	(195,075; 195,609)	0.11	0.60	0.57
SOPTUB	194,372	(193,859; 194,884)	−0.39	1.2	0.58
BINUB	195,162	(194,665; 195,658)	0.00	0.76	0.75

Figure 8. Average overbooking levels in the last period and number of show-ups: Example 3, Scenario 1.

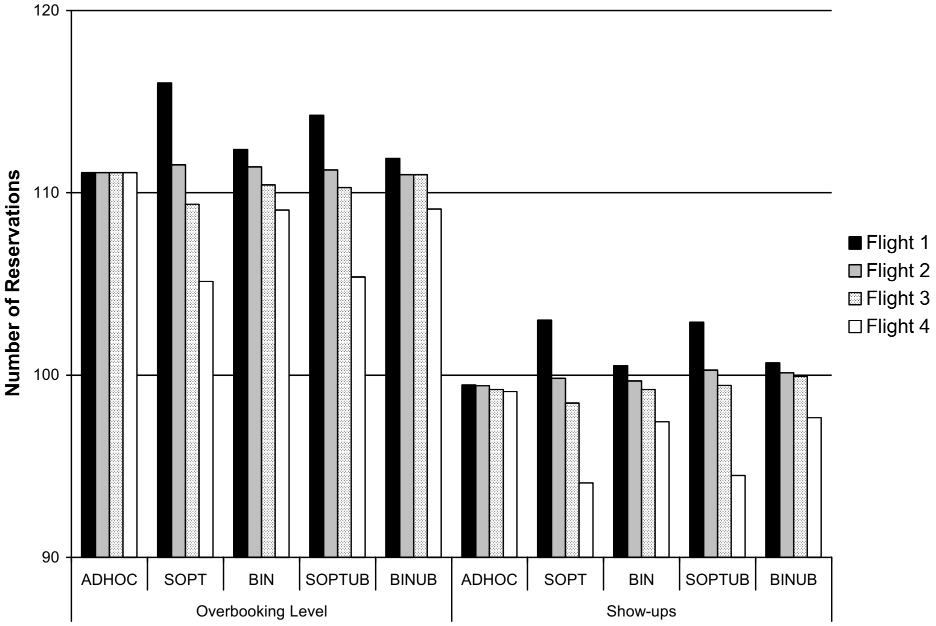


Table 5. Average revenues and service levels for different policies in Example 3, Scenario 2.

	Revenue			Service Levels	
Policy	Avg.	90% Conf. Int.	%Over ADHOC	%Delayed	%Denied
ADHOC	185,639	(185,202; 186,077)	0.0	0.47	0.42
SOPT	186,000	(185,306; 186,694)	0.19	1.33	0.03
BIN	186,389	(185,719; 187,059)	0.40	0.67	0.2
SOPTUB	188,176	(187,467; 188,884)	1.37	4.09	0.14
BINUB	188,328	(187,494; 189,161)	1.45	4.49	0.66

Figure 9. Average overbooking levels in the last period and number of show-ups: Example 3, Scenario 2.

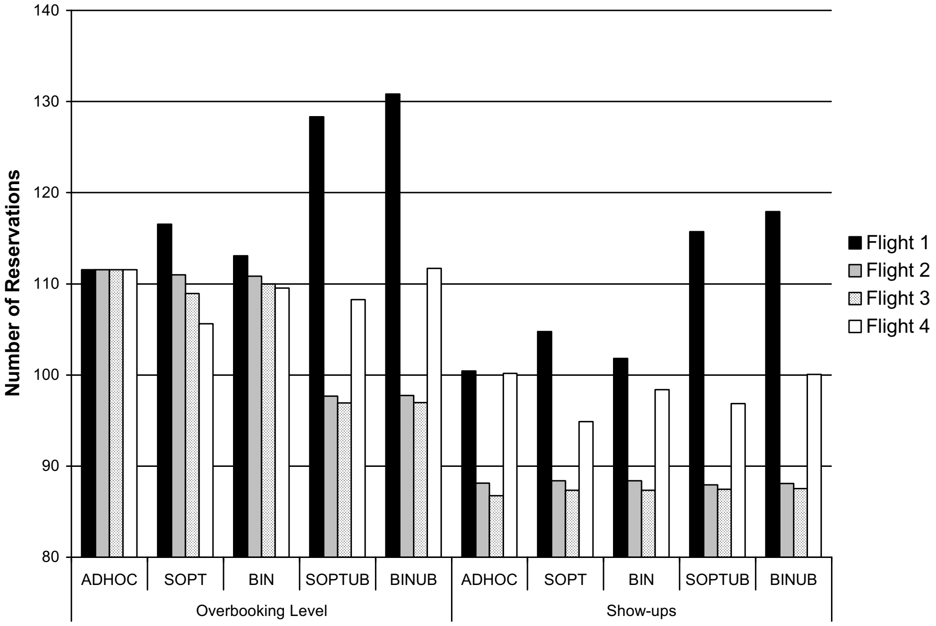
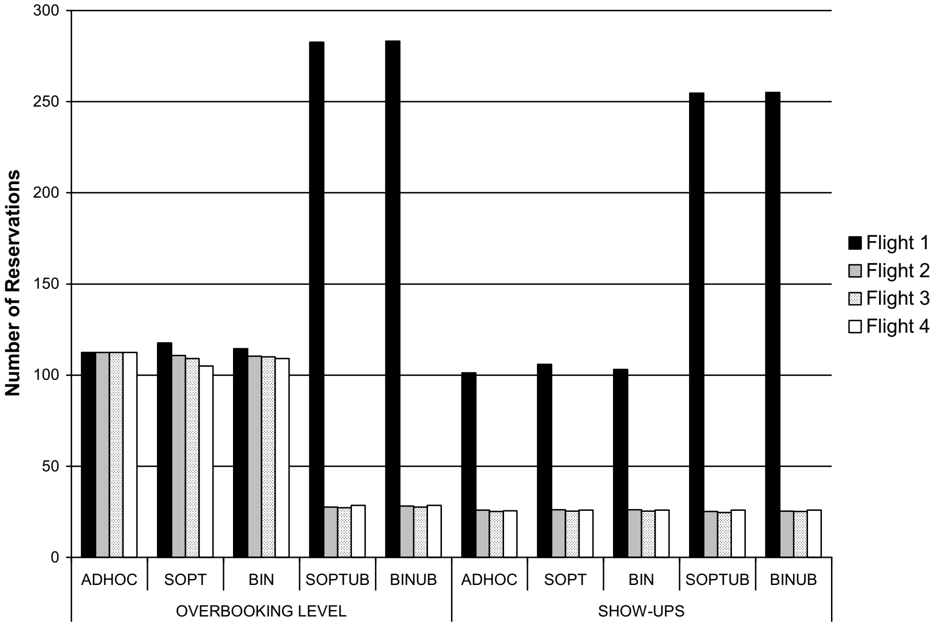


Figure 10. Average overbooking levels in the last period and number of show-ups: Example 3, Scenario 3.



imposing an exogeneous upper bound on the overbooking level of Flight 1. Still, it does illustrate the type of behavior that can occur when substitution options are incorporated into the reservation acceptance decision.

6. Conclusions

We have presented a two-period optimization model to determine joint overbooking levels for multiple-class revenue management settings where substitution among classes is allowed as a recourse option. We analyzed the structural properties of the problem and proposed a solution procedure that can be used within a dynamic decision-making process. The properties and the solution procedure are valid for general substitution rules (involving upgrades, downgrades, or both). Numerical results from several examples show that in some cases accounting for substitution when setting overbooking levels can significantly increase revenues net of penalties, even when compared to the ad hoc use of substitution. This suggests there is potential to improve overbooking practices for adjacent flights, multicabin flights, car rental fleets, multiroom hotels, and other revenue management applications where substitution options are prevalent and widely used in an ad hoc fashion.

At the same time, our approach only represents a first attempt at addressing these issues. Our model is an approximation of the true dynamic problem. Explicitly accounting for the dynamic of arrivals and cancellations would be desirable. In addition, assignments of customers to inventory classes may need to be performed prior to observing the realization of all cancellations, unlike the perfect-information assignment in our model. Some reasonable approach to this sequential assignment problem is another worthwhile topic for future research.

Appendix

A.1. Finding the Difference Function or the Gradient of the Service Period Function

Here is an example with binomial distribution to show the difference function for an expected value. For simplicity, we use $n = 1$. Let u be a nonnegative integer and $Z(u) \sim$

Table 6. Average revenues and service levels for the policies in Example 3, Scenario 3.

	Revenue			Service Levels	
Policy	Avg.	90% Conf. Int.	%Over ADHOC	%Delayed	%Denied
ADHOC	88,234	(87,900; 88,567)	0.0	1.22	0.0
SOPT	89,381	(89,312; 90,350)	1.30	3.2	0.0
BIN	89,192	(88,668; 89,715)	0.40	1.08	0.0
SOPTUB	109,820	(109,256; 110,383)	24.46	46	0.0
BINUB	110,148	(109,557; 110,738)	24.84	46	0.0

binomial(p, u). Then, we know

$$P(Z(u+1)=0)-P(Z(u)=0)=-pP(Z(u)=0), \quad (15)$$

$$P(Z(u+1)=u+1)=pP(Z(u)=u), \quad (16)$$

and for $k=1,\ldots,u$,

$$\begin{aligned}&P(Z(u+1)=k)-P(Z(u)=k)\\&\quad=p[P(Z(u)=k-1)-P(Z(u)=k)].\end{aligned} \quad (17)$$

Suppose we are interested in the first-order difference function of $E[V(Z(u))]$. Then,

$$\begin{aligned}&\Delta E[V(Z(u))]\\&\quad=E[V(Z(u+1))]-E[V(Z(u))]\\&\quad=\sum_{z=0}^{u+1}V(z)P(Z(u+1)=z)\\&\qquad-\sum_{z=0}^{u}V(z)P(Z(u)=z)\\&\quad=V(0)[P(Z(u+1)=0)-P(Z(u)=0)]\\&\qquad+\sum_{k=1}^{u}V(k)[P(Z(u+1)=k)-P(Z(u)=k)]\\&\qquad+V(u+1)P(Z(u+1)=u+1)\\&\quad=-pV(0)P(Z(u)=0)+\sum_{k=1}^{u}pV(k)\\&\qquad\cdot[P(Z(u)=k-1)-P(Z(u)=k)]\\&\qquad+pV(u+1)P(Z(u)=u).\end{aligned} \quad (18)$$

Rearranging the terms in (18) using Equations (15), (16), and (17), we get

$$\Delta E[V(Z(u))]=\sum_{k=0}^{u}p[V(k+1)-V(k)]P(Z(u)=k).$$

If we have $Z(u)\sim\text{Poisson}(pu)$, u a nonnegative real number, then we have a similar result. The Poisson probabilities satisfy the following:

$$\frac{\partial}{\partial u}P(Z(u)=0)=-pP(Z(u)=0), \quad (19)$$

$$\frac{\partial}{\partial u}P(Z(u)=k)=p[P(Z(u)=k-1)-P(Z(u)=k)] \quad \text{for } k\geqslant 1. \quad (20)$$

Suppose there exists an integrable (with respect to Lebesgue measure) function $h(z)$ such that

$$\left|V(z)\frac{\partial}{\partial u}P(Z(u)=z)\right|\leqslant h(z).$$

Then, the gradient of $E[V(Z(u))]$ becomes

$$\begin{aligned}&\frac{d}{du}E[V(u)]\\&\quad=\frac{d}{du}\sum_{z=0}^{\infty}V(z)P(Z(u)=z)\\&\quad=\sum_{z=0}^{\infty}V(z)\frac{d}{du}P(Z(u)=z)\\&\qquad-pV(0)P(Z(u)=0)\\&\qquad+\sum_{z=1}^{\infty}V(z)p[P(Z(u)=z-1)-P(Z(u)=z)].\end{aligned} \quad (21)$$

Rearranging the terms in (21) using Equations (19) and (20), we get

$$\frac{d}{du}E[V(u)]=\sum_{z=0}^{\infty}p[V(z+1)-V(z)]P(Z(u)=z).$$

A.2. Proof of Lemma 3

From our analysis of the DG estimate, the partial derivatives of $G(u)$ can be expressed as

$$\begin{aligned}\frac{\partial}{\partial u_i}G(u)&=r_i-q_i(1-p_i)\\&\quad+p_iE[V_0(Z(u)+e_i)-V_0(Z(u))].\end{aligned}$$

By the submodularity of V_0, we have $V_0(z+e_i)-V_0(z)\leqslant V_0(e_i)-V_0(0)$, which is finite as the allocation penalties in the service period (coefficients a_{ij}) are finite. Hence, $(\partial/\partial u_i)G(u)$ is finite. Recall

$$\begin{aligned}\frac{\partial}{\partial u_i}G(u)&=r_i-q_i(1-p_i)\\&\quad+\sum_{z_1=0}^{\infty}\cdots\sum_{z_n=0}^{\infty}p_i[V_0(z+e_i)-V_0(z)]\\&\quad\cdot P(Z_1(u_1)=z_1)\cdots P(Z_n(u_n)=z_n).\end{aligned}$$

The decision variables, u_i, appear only in the last term with

$$P(Z_i(u_i)=z_i)=e^{-p_iu_i}\frac{(p_iu_i)^{z_i}}{z_i!}.$$

Thus, the last term in the above expression is continuous with respect to u. Therefore, $G(u)$ is continuously differentiable.

For the second derivatives, we have

$$\begin{aligned}&\frac{\partial}{\partial u_j\partial u_i}G(u)\\&\quad=\frac{\partial}{\partial u_j}[r_i-q_i(1-p_i)+p_iE[V_0(Z(u)+e_i)-V_0(Z(u))]]\\&\quad=p_i\frac{\partial}{\partial u_j}E[V_0(Z(u)+e_i)]-p_i\frac{\partial}{\partial u_j}E[V_0(Z(u))]\\&\quad=p_ip_jE[V_0(Z(u)+e_i+e_j)-V_0(Z(u)+e_i)]\\&\qquad-p_ip_jE[V_0(Z(u)+e_j)-V_0(Z(u))].\end{aligned}$$

By the same argument as above, the function G is twice continuously differentiable.

A.3. Proof of Lemma 4

We consider $\mathrm{Var}(D_i^k)$ at each iteration k of the algorithm, when u^k is given. Using (10), we have

$$\begin{aligned}
\mathrm{Var}(D_i^k) &= \mathrm{Var}(H_i(Z(u^k)))\\
&= p_i^2 \mathrm{Var}(V_0(Z(u^k)+e_i) - V_0(Z(u^k)))\\
&\leqslant E[(V_0(Z(u^k)+e_i) - V_0(Z(u^k)))^2]\\
&\leqslant \sum_{z_1=0}^{\infty} \cdots \sum_{z_n=0}^{\infty} (V_0(z+e_i) - V_0(z))^2\\
&\qquad \cdot P(Z_1(u_1^k)=z_1) \cdots P(Z_n(u_n^k)=z_n)\\
&\leqslant (V_0(0+e_i) - V_0(0))^2\\
&\qquad \cdot \sum_{z_1=0}^{\infty} \cdots \sum_{z_n=0}^{\infty} P(Z_1(u_1^k)=z_1) \cdots P(Z_n(u_n^k)=z_n)\\
&= V_0(e_i)^2 = \Big(\max_{j=0,1,\ldots,m} a_{ij}\Big)^2 < C < \infty.
\end{aligned}$$

The above result follows from submodularity of V_0; i.e., $V_0(z+e_i) - V_0(z) \leqslant V_0(0+e_i) - V_0(0)$ for all $z \geqslant 0$.

Now to show the same result for SFG, using (12), we have

$$\begin{aligned}
\mathrm{Var}(D_i^k) &= \mathrm{Var}(H_i(Z(u^k)))\\
&= \mathrm{Var}((Z_i(u_i^k)/u_i^k - p_i)\ V_0(Z(u^k)))\\
&\leqslant E[(Z_i(u_i^k)/u_i^k - p_i)^2 V_0(Z(u^k))^2]\\
&= \sum_{z_1=0}^{\infty} \cdots \sum_{z_n=0}^{\infty} (z_i/u_i^k - p_i)^2 V_0(z)^2\\
&\qquad \cdot P(Z_1(u_1^k)=z_1) \cdots P(Z_n(u_n^k)=z_n). \qquad (22)
\end{aligned}$$

To prove that the last expression is finite, we look at the properties of V_0 and the moments of Poisson-distributed random variables. We know $-a^*(z_1 + \cdots + z_n) \leqslant V_0(z) \leqslant V_0^* < \infty$, where $a^* > 0$ is the highest overbooking cost ($a^* < \infty$), and V_0^* is the highest revenue we can get from service allocation. Because the cost/revenue parameters and the capacities are finite, and the service allocation problem is always feasible, there exists a $V_0^* < \infty$. Because $Z_i(u_i)$ has finite first and second moments, we have the following:

$$\begin{aligned}
&\sum_{z_1=0}^{\infty} \cdots \sum_{z_n=0}^{\infty} (z_i/u_i^k - p_i)^2 (V_0^*)^2\\
&\quad \cdot P(Z_1(u_1^k)=z_1) \cdots P(Z_n(u_n^k)=z_n) < \infty. \qquad (23)
\end{aligned}$$

Similarly, we have

$$\begin{aligned}
&\sum_{z_1=0}^{\infty} \cdots \sum_{z_n=0}^{\infty} (z_i/u_i^k - p_i)^2 (a^*)^2 (z_1 + \cdots + z_n)^2\\
&\quad \cdot P(Z_1(u_1^k)=z_1) \cdots P(Z_n(u_n^k)=z_n) < \infty. \qquad (24)
\end{aligned}$$

The above expression holds because $Z_i(u_i)$ for $i = 1, \ldots, n$ have finite kth moments for $k = 1, \ldots, 4$ (this is true for $k \geqslant 4$, but we only need $k = 4$ at most). Therefore, we can find a finite constant C that bounds the variance of the estimate. (For example, C can be the maximum of the expressions in (23) or (24) that is greater than (22).)

References

Alstrup, J., S. Boas, O. B. G. Madsen, R. V. V. Vidal. 1986. Booking policy for flights with two types of passengers. *Eur. J. Oper. Res.* **27** 274–288.

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

Birge, R. J., F. Louveaux. 1997. *Introduction to Stochastic Programming*. Springer Series in Operations Research. Springer-Verlag, New York.

Bitran, G. R., S. Dasu. 1992. Ordering policies in an environment of stochastic yields and substitutable demands. *Oper. Res.* **40** 999–1017.

Bitran, G. R., S. Gilbert. 1994. Co-production processes with random yields in the semiconductor industry. *Oper. Res.* **42** 476–491.

Bitran, G. R., S. Gilbert. 1996. Managing hotel reservations with uncertain arrivals. *Oper. Res.* **44** 35–49.

Carroll, W. J., R. C. Grimes. 1995. Evolutionary change in product management: Experiences in the car rental industry. *Interfaces* **25** 84–104.

Chatwin, R. E. 1992. Optimal airline overbooking. Ph.D. thesis, Stanford University, Palo Alto, CA.

Chatwin, R. E. 1999. Continuous-time airline overbooking with time dependent fares and refunds. *Transportation Sci.* **33** 182–191.

Chen, J. 1997. Substitution and inspection models in production-inventory systems. Ph.D. thesis, Department of IEOR, Columbia University, New York.

Edelstein, M., M. Melnyk. 1977. The pool control system. *Interfaces* **8** 21–36.

Erlenkotter, D. 1970. Preinvestment planning for capacity expansion: A multi-location dynamic model. Ph.D. thesis, Stanford University, Palo Alto, CA.

Geraghty, M. K., E. Johnson. 1996. Revenue management saves National Car Rental. *Interfaces* **27** 107–127.

Hsu, A., Y. Bassok. 1999. Random yield and random demand in a production system with downward substitution. *Oper. Res.* **47** 277–290.

Jones, C. P., T. J. Lowe, G. Muller, X. Ning, Y. Yinyu, J. L. Zydiak. 1995. Specially structured uncapacitated facility location problems. *Oper. Res.* **43** 661–669.

Karaesmen, I. 2001. Three essays on revenue management. Unpublished Ph.D. thesis, Columbia University, New York.

Kimes, S. E. 1989. Yield management: A tool for capacity-constrained service firms. *J. Oper. Management* **8** 348–363.

Kushner, H. J., D. S. Clark. 1978. *Stochastic Approximation Methods for Constrained and Unconstrained Systems*. Applied Mathematical Sciences Series, Vol. 26. Springer-Verlag, New York.

Ladany, S. P. 1976. Dynamic operating rules for motel reservations. *Decision Sci.* **7** 829–841.

Liberman, V., U. Yechiali. 1978. On the hotel overbooking problem—An inventory system with stochastic cancellations. *Management Sci.* **24** 1117–1126.

Martinez, R., M. Sanchez. 1970. Automatic booking level control. *Proc. Tenth AGIFORS Sympos.* American Airlines, New York.

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

Ratliff, R. M. 1998. Ideas on overbooking. Presentation at AGIFORS Reservation and Yield Management Study Group Meeting, Melbourne, Australia.

Robinson, L. W. 1990. Optimal and approximate policies in multiperiod, multilocation inventory models with transshipments. *Oper. Res.* **38** 278–295.

Ross, S. 1996. *Stochastic Processes*, 2nd ed. John Wiley and Sons, New York.

Rothstein, M. 1971. An airline overbooking model. *Transportation Sci.* **5** 180–192.

Rothstein, M. 1974. Hotel overbooking as a Markovian sequential decision process. *Decision Sci.* **5** 389–404.

Rothstein, M. 1985. Operations research and the airline overbooking problem. *Oper. Res.* **33** 237–248.

Rothstein, M., A. W. Stone. 1967. Passenger booking levels. *Proc. Seventh AGIFORS Sympos.* American Airlines, New York.

Rubinstein, R. Y., A. Shapiro. 1993. *Discrete Event Systems: Sensitivity Analysis and Stochastic Optimization by the Score Function Method.* Wiley, New York.

Ruszczynski, A., W. Syski. 1986. A method of aggregate stochastic subgradients with on-line stepsize rules for convex stochastic programming problems. *Math. Programming Stud.* **28** 113–131.

Samuelson, P. A. 1952. Spatial price equilibrium and linear programming. *Amer. Econom. Rev.* **42** 283–303.

Shaked, M., J. G. Shanthikumar. 1988. Stochastic convexity and its applications. *Adv. Appl. Probab.* **20** 427–446.

Shapley, L. S. 1962. Complements and substitutes in the optimal assignment problem. *Naval Res. Logist. Quart.* **9** 45–48.

Shlifer, E., Y. Vardi. 1975. An airline overbooking policy. *Transportation Sci.* **9** 101–114.

Smith, B., J. Leimkuhler, R. Darrow, J. Samuels. 1992. Yield management at American Airlines. *Interfaces* **22** 8–31.

Subramanian, J., S. Stidham Jr., C. Lautenbacher. 1999. Airline yield management with overbooking, cancellations and no-shows. *Transportation Sci.* **33** 136–146.

Sundaram, R. K. 1996. *A First Course in Optimization Theory*. Cambridge University Press, New York.

Taylor, C. J. 1962. The determination of passenger booking levels. *Proc. Second AGIFORS Sympos.* American Airlines, New York.

Thompson, H. R. 1961. Statistical problems in airline reservation control. *Oper. Res. Quart.* **12** 167–185.

Topkis, D. M. 1998. *Supermodularity and Complementarity*. Princeton University Press, Princeton, NJ.

Weatherford, L. R., S. E. Bodily. 1992. A taxonomy and research overview of perishable-asset revenue management: Yield management, overbooking and pricing. *Oper. Res.* **40** 831–844.

Yano, C. A., H. L. Lee. 1995. Lot sizing with random yields: A review. *Oper. Res.* **43** 311–335.