A Stochastic Approximation Method for the Revenue Management Problem on a Single Flight Leg with Discrete Demand Distributions

Sumit Kunnumkal
School of Operations Research and Industrial Engineering, Cornell University, Ithaca, New York 14853, USA
sumit@orie.cornell.edu

Huseyin Topaloglu
School of Operations Research and Industrial Engineering, Cornell University, Ithaca, New York 14853, USA
topaloglu@orie.cornell.edu

February 16, 2007

Abstract

We consider the problem of optimally allocating the seats on a single flight leg to the demands from multiple fare classes that arrive sequentially. It is well-known that the optimal policy for this classical problem is characterized by a set of protection levels. In this paper, we propose a stochastic approximation method to compute the optimal protection levels under the assumption that the demand distributions are not known and we only have access to the samples from the demand distributions. The novel aspect of our method is that it works with the nonsmooth version of the problem where the capacity can only be allocated in integer quantities. We show that our method obtains valid stochastic subgradients of the value functions without computing expectations explicitly and discuss applications to the case where the demand information is censored by the seat availability. Numerical experiments indicate that our method is especially advantageous when the total expected demand exceeds the capacity by a significant margin.
A recurrent problem in the revenue management literature involves optimally allocating the seats on a single flight leg to the demands from multiple fare classes that arrive sequentially. Given the demand from the current fare class and the number of unsold seats, the decision that needs to be made is how many seats to sell to the current fare class. Starting with Littlewood (1972), this problem has been studied extensively and it is well-known that the optimal policy is characterized by one protection level for each fare class. Specifically, letting $n$ be the number of fare classes, there exists a set of protection levels $\{y^*_j : j = 1, \ldots, n\}$ such that it is optimal to keep the number of unsold seats just after making the decisions for fare class $j$ as close as possible to $y^*_j$. In other words, letting $x_j$ be the number of unsold seats just before making the decisions for fare class $j$ and $[\cdot]^+ = \max\{0, \cdot\}$, it is optimal to make $[x_j - y^*_j]^+$ seats available for sale to fare class $j$. If the demand from fare class $j$ does not exceed $[x_j - y^*_j]^+$, then all of the demand is satisfied. Otherwise, only $[x_j - y^*_j]^+$ seats are sold. This structure of the optimal policy arises from the fact that the value functions in the dynamic programming formulation of the problem are concave in the number of unsold seats. In this case, the computation of the optimal protection levels through the Bellman equations requires solving a number of convex optimization problems, which is a relatively simple task as long as the demand distributions are known.

In this paper, we propose a stochastic approximation method to compute the optimal protection levels when the demand distributions are not known and we only have access to the samples from the demand distributions. We work with a particular version of the problem where the demand distributions are discrete and the fare classes that generate lower revenues arrive earlier than the fare classes that generate higher revenues. We develop a novel method that uses the dynamic programming formulation of the problem in conjunction with the samples from the demand distributions to approximate the stochastic subgradients of the value functions. By showing that our approximate stochastic subgradients are indeed accurate in the limit, we establish that the iterates of our stochastic approximation method converge to a set of optimal protection levels with probability one (w.p.1). To deal with the case where the demand information is censored by the seat availability, we provide alternative versions of our method that remain applicable when we can only observe the number of seats sold to a fare class, but not necessarily the amount of demand from a fare class.

Although there has been work on using stochastic approximation methods to compute the optimal protection levels, our paper makes unique contributions. Brumelle and McGill (1993) characterize the conditions that should be satisfied by the optimal protection levels and van Ryzin and McGill (2000) exploit these conditions to develop a stochastic approximation method. However, this method is tightly related to the optimality conditions in Brumelle and McGill (1993) and it is not clear whether it can be extended to another problem class. In contrast, we work with the dynamic programming formulation of the problem and it is possible to extend our method to inventory control problems where the value functions are convex and the base stock policies are optimal. Furthermore, the step directions used by our method are related to the stochastic subgradients of the value functions, whereas this is not the case for the method proposed by van Ryzin and McGill (2000). Stochastic subgradients of the value functions can be particularly useful when making tactical decisions such as setting the capacity.
Figure 1: Plot of $\{R(y_1,0) : y_1 = 0, \ldots, 8\}$. The problem parameters are $c = 5$, $r_1 = 5$, $r_2 = 10$. The demands from the two fare classes are deterministic and we have $D_1 = 3$, $D_2 = 3$ w.p.1.

of the flight leg. Huh and Rusmevichientong (2006) also propose a stochastic approximation method to compute the optimal protection levels. There are similarities between their method and ours as both exploit the dynamic programming formulation of the problem, but Huh and Rusmevichientong (2006) use the results from the online convex optimization literature pioneered by Zinkevich (2003), whereas we use the stochastic approximation theory. Finally, both van Ryzin and McGill (2000) and Huh and Rusmevichientong (2006) work with continuous demand distributions. To our knowledge, our method is the only one that works with discrete demand distributions and has a convergence guarantee for the performance of the policy. To deal with discrete demand distributions, van Ryzin and McGill (2000) propose a randomized version of their method. The iterates of this version converge to a set of optimal protection levels, but the randomization results in suboptimality for the performance of the policy.

It is also important to note that the total expected revenue for the seat allocation problem is not concave when viewed as a function of the protection levels. To illustrate, assuming that there are two fare classes, and using $c$ to denote the initial capacity and $\{r_1, r_2\}$, $\{D_1, D_2\}$ and $\{y_1, y_2\}$ to respectively denote the revenues, demand random variables and protection levels, the total expected revenue is

$$R(y_1, y_2) = r_1 \mathbb{E}\left\{ \min\{[c - y_1]^+, D_1\} \right\} + r_2 \mathbb{E}\left\{ \min\left\{\left[c - \min\{[c - y_1]^+, D_1\} - y_2\right]^+, D_2\right\} \right\},$$

where we use the fact that if there are $x_j$ unsold seats just before making the decisions for fare class $j$, then we make $[x_j - y_j]^+$ seats available for sale to fare class $j$ and sell $\min\{[x_j - y_j]^+, D_j\}$ seats. Figure 1 plots a cross section of $R(\cdot, \cdot)$ for a problem instance and shows that this function may not be concave. Consequently, if we naively attempt to compute the optimal protection levels by solving the problem $\max_{(y_1,y_2)} R(y_1,y_2)$ through a stochastic approximation method, then we do not necessarily obtain the optimal protection levels. Our results in this paper, however, show that it is possible to develop a stochastic approximation method to compute the optimal protection levels as long as we use step directions that are based on the dynamic programming formulation of the problem.

Since we work with discrete demand distributions, our stochastic approximation method has some drawbacks when the demand information is censored by the seat availability. Specifically, if such demand censorship occurs and we sell all of the seats that we make available for sale to a fare class, then we only know that the demand is greater than or equal to the seat availability. However, our method requires knowing whether the demand is strictly greater than the seat availability. If a somewhat relaxed view of
demand censorship is possible and we can indeed observe whether the demand strictly exceeds the seat availability, then our method remains applicable with no modifications. This relaxed view of demand censorship is essentially equivalent to assuming that we can observe the demand from the first customer that we turn down. To address the case where the demand information is censored and the relaxed view of demand censorship is not possible, we also provide two alternative versions of our method. For the first alternative version, we show that the distance between its iterates and the optimal protection levels is bounded by the number of fare classes in the limit w.p.1. The second alternative version is a heuristic modification of the first one with somewhat more desirable practical performance but no convergence guarantee. We emphasize that the drawbacks mentioned in this paragraph arise solely due to the fact that we work with discrete demand distributions. When working with continuous demand distributions, the event that the demand is equal to the seat availability occurs with probability zero. In this case, there is essentially no distinction between knowing that the demand is greater than or equal to or strictly greater than the seat availability.

The rest of the paper is organized as follows. Section 2 briefly reviews the other related literature. Section 3 gives a dynamic formulation of the seat allocation problem. Section 4 describes our stochastic approximation method and Section 5 proves its convergence. Section 6 considers the case where the demand information is censored by the seat availability. Section 7 provides numerical experiments.

2 Review of Other Related Literature

There is extensive literature on the seat allocation problem on a single flight leg. Although it makes a number of restrictive assumptions and the airlines almost invariably operate hub-and-spoke networks, the problem has important practical implications. For example, it justifies, at least to a certain extent, the use of protection level policies for complex networks. Furthermore, there are a variety of techniques to decompose the revenue management problem over a network into a sequence of single flight legs. Most of the literature on the single flight leg problem assumes that there are multiple fare classes and the demands from different fare classes occur over nonoverlapping time intervals. This ensures that we can formulate the problem as a dynamic program with the number of stages being equal to the number of fare classes. Motivated by the fact that leisure travelers tend to book earlier than the business travelers, it is also a common assumption that the fare classes that generate lower revenues arrive earlier than the fare classes that generate higher revenues. An interesting consequence of this second assumption is that the optimal protection levels are nested. That is, the optimal protection level for a fare class is greater than the optimal protection levels for the fare classes that arrive later.

Littlewood (1972), Curry (1990), Wollmer (1992) and Brumelle and McGill (1993) employ the two assumptions in the previous paragraph and show the optimality of protection level policies. Robinson (1995) characterizes the structure of the optimal policy under the assumption that the demands from different fare classes occur over nonoverlapping time intervals, but the fare classes that generate lower revenues do not necessarily arrive earlier. Lee and Hersh (1993) and Lautenbacher and Stidham (1999) focus on the single leg problem when the demands from different fare classes do not necessarily arrive over nonoverlapping time intervals. In the latter two cases, it is still possible to show that a variation of
protection level policies is optimal. We refer the reader to Talluri and van Ryzin (2004) for a coverage of the related revenue management literature.

The use of stochastic approximation methods for solving stochastic optimization problems is well-known. Kushner and Clark (1978), Ermoliev (1988) and Bertsekas and Tsitsiklis (1996) give a coverage of the theory of stochastic approximation methods. There are numerous papers that use these methods for solving the revenue management problem over a network. In particular, van Ryzin and Vulcano (2004), Bertsimas and de Boer (2005) and van Ryzin and Vulcano (2006) describe methods to compute protection levels, Topaloglu (2007) describes a method to compute bid prices and Karaesmen and van Ryzin (2004) describe a method to compute overbooking limits. As far as other application areas are concerned, L’Ecuyer and Glynn (1994), Fu (1994), Glasserman and Tayur (1995), Bashyam and Fu (1998) and Mahajan and van Ryzin (2001) focus on queueing and inventory control. Kunnumkal and Topaloglu (2006) show that a method similar to ours can be used to compute the optimal base stock levels in inventory control problems. However, since they work with continuous demand distributions, their proof technique is considerably different from ours.

3 Problem Formulation

We want to use \(c\) seats available on a single flight leg to satisfy the demands from \(n\) fare classes that arrive sequentially. We index the fare classes such that the demand from fare class 1 arrives first and the demand from fare class \(n\) arrives last. If we sell a seat to fare class \(j\), then we generate a revenue of \(r_j\). We assume that the revenues satisfy \(0 < r_1 \leq r_2 \leq \ldots \leq r_n\) so that the demands from the cheaper fare classes arrive earlier. The demands from different fare classes are random and we use \(D_j\) to denote the demand from fare class \(j\). We assume that \(D_j\) is a positive and integer random variable and \(\{D_j : j = 1, \ldots, n\}\) are independent of each other. We are interested in maximizing the total expected revenue from \(n\) fare classes.

If we use \(x_j\) to denote the remaining capacity just before making the decisions for fare class \(j\), \(u_j\) to denote the number of seats sold to fare class \(j\) and \(d_j\) to denote a particular realization of \(D_j\), then the optimal policy can be found by solving the optimality equations

\[
v_j(x_j, d_j) = \max_{0 \leq u_j \leq \min\{x_j, d_j\}} r_j u_j + \mathbb{E}\{v_{j+1}(x_j - u_j, D_{j+1})\}, \tag{2}
\]

with \(v_{n+1}(\cdot, \cdot) = 0\). The constraints in the problem above ensure that the number of seats sold do not exceed the remaining capacity and the demand from fare class \(j\). Alternatively, if we let \(y_j = x_j - u_j\) be the remaining capacity just after making the decisions for fare class \(j\), then (2) can be written as

\[
v_j(x_j, d_j) = \left\{ \max_{[x_j-d_j]^+ \leq y_j \leq x_j} -r_j y_j + \mathbb{E}\{v_{j+1}(y_j, D_{j+1})\} \right\} + r_j x_j, \tag{3}
\]

where the constraints follow from the fact that \(x_j - \min\{x_j, d_j\} = \max\{0, x_j - d_j\}\). It is possible to show that \(\{v_j(\cdot, D_j) : j = 1, \ldots, n\}\) are piecewise-linear concave functions with points of nondifferentiability being a subset of integers for all realizations of \(\{D_j : j = 1, \ldots, n\}\). In this case, it is easy to show that the optimal policy is characterized by a set of protection levels \(\{y_j^* : j = 1, \ldots, n\}\), where \(y_j^*\) can be
computed as a maximizer of the function

\[ f_j(y_j) = -r_j y_j + \mathbb{E}\{v_{j+1}(y_j, D_{j+1})\} \]  

(4)

over the interval \([0, c]\). This is to say that if the remaining capacity just before making the decisions for fare class \(j\) is \(x_j\) and the demand from fare class \(j\) is \(d_j\), then it is optimal to sell \(\min\{[x_j - y_j^*]^+, d_j\}\) seats to fare class \(j\). The protection level terminology is due to the fact that it is optimal to protect \(y_j^*\) seats for the demand from fare classes \(\{j+1, \ldots, n\}\) when making the decisions for fare class \(j\).

Since the demands from the cheaper fare classes arrive earlier, it is also possible to show that the optimal protection levels are nested. In other words, the optimal number of seats to protect for the demand from fare classes \(\{j, \ldots, n\}\) is at least as large as the optimal number of seats to protect for the demand from fare classes \(\{j+1, \ldots, n\}\). To state this mathematically, we let

\[ \mathcal{Y}_j^* = \arg\max_{0 \leq y_j \leq c} f_j(y_j). \]  

(5)

Therefore, we can use any element of \(\mathcal{Y}_j^*\) as the optimal protection level when making the decisions for fare class \(j\). The fact that the optimal protection levels are nested implies that \(\min_{y_j \in \mathcal{Y}_j^*} y_j \geq \min_{y_{j+1} \in \mathcal{Y}_{j+1}^*} y_{j+1} \) and \(\max_{y_j \in \mathcal{Y}_j^*} y_j \geq \max_{y_{j+1} \in \mathcal{Y}_{j+1}^*} y_{j+1}\) for all \(j = 1, \ldots, n - 1\). In this case, we can choose \(y_1^* \in \mathcal{Y}_1^*\), \(y_2^* \in \mathcal{Y}_2^*\), \ldots, \(y_n^* \in \mathcal{Y}_n^*\) such that \(y_1^* \geq y_2^* \geq \ldots \geq y_n^*\).

Our dynamic programming formulation differs from the existing literature in two aspects. First, we index the fare classes such that fare classes \(1\) and \(n\) respectively correspond to the cheapest and most expensive fare classes, whereas the existing literature usually indexes the fare classes in the reverse order. The motivation for our choice is that it is common to refer to a cheaper fare class as a lower fare class and it is more consistent to index a cheaper fare class with a smaller integer. Second, we use a two-dimensional state variable in (2) and (3), whereas the existing literature usually uses a one-dimensional state variable. It is possible to use a one-dimensional state variable in (2) and (3) by simply letting \(\hat{v}_j(x_j) = \mathbb{E}\{v_j(x_j, D_j)\}\) and taking the expectations of both sides. However, our dynamic programming formulation will be more useful for the subsequent development in the paper. Lastly, we note that all of the results that we mention in this section are quite standard and the details can be found in Brumelle and McGill (1993) and Talluri and van Ryzin (2004).

4 Stochastic Approximation Method

In this section, we consider computing the optimal protection levels by using a stochastic approximation method. By (4), we can compute a stochastic subgradient of \(f_j(\cdot)\) at \(y_j\) through

\[ \Delta_j(y_j, d_{j+1}) = -r_j + \hat{v}_{j+1}(y_j, d_{j+1}), \]  

(6)

where we use \(\hat{v}_{j+1}(y_j, d_{j+1})\) to denote a stochastic subgradient of \(\mathbb{E}\{v_{j+1}(\cdot, D_{j+1})\}\) at \(y_j\). In other words, if we use \(\partial v_{j+1}(y_j, d_{j+1})\) to denote the subdifferential of \(v_{j+1}(\cdot, d_{j+1})\) at \(y_j\), then we have \(\hat{v}_{j+1}(y_j, d_{j+1}) \in \partial v_{j+1}(y_j, d_{j+1})\). Interchanging the orders of all expectations and subgradients throughout the paper trivially follows from the fact that the demand distributions are discrete and the capacity is finite. In
this case, letting \( \{ y_j^k : j = 1, \ldots, n \} \) be the estimates of the optimal protection levels at iteration \( k \), \( \{ D_j^k : k = 1, \ldots, n \} \) be the demand random variables at iteration \( k \) and \( \{ \alpha_j^k : j = 1, \ldots, n \} \) be a sequence of step size parameters, we can update our estimates of the optimal protection levels by

\[
y_j^{k+1} = \min \{ [ y_j^k + \alpha_j^k \Delta_j(y_j^k, D_{j+1}^k)]^+, c \},
\]

(7)

where the operator \( \min \{ [\cdot]^+, c \} \) ensures that the estimates of the optimal protection levels always lie in the interval \([0, c]\). If the sequence of protection levels \( \{ y_j^k : j = 1, \ldots, n \} \) is generated by (7), then we can use the standard results on stochastic approximation methods to show that \( \{ y_j^k : j = 1, \ldots, n \} \) converges to a set of optimal protection levels w.p.1. However, this approach is clearly not realistic because the computation in (6) requires the knowledge of \( \{ v_j(\cdot) : j = 1, \ldots, n \} \). The stochastic approximation method that we propose in this section is based on constructing tractable approximations to the stochastic subgradients of \( \{ f_j(\cdot) : j = 1, \ldots, n \} \).

Since \( f_j(\cdot) \) is concave and the optimal protection level \( y_j^* \) is a maximizer of this function over the interval \([0, c]\), we can write (3) as

\[
v_j(x_j, d_j) = \begin{cases} -r_j [x_j - d_j]^+ + r_j x_j + \mathbb{E}\{ v_{j+1}(x_j - d_j^+, D_{j+1}) \} & \text{if } y_j^* < [x_j - d_j]^+ \\ -r_j y_j^* + r_j x_j + \mathbb{E}\{ v_{j+1}(y_j^*, D_{j+1}) \} & \text{if } [x_j - d_j]^+ \leq y_j^* \leq x_j \\ \mathbb{E}\{ v_{j+1}(x_j, D_{j+1}) \} & \text{if } x_j < y_j^* \end{cases}
\]

(8)

for \( x_j \in [0, c] \). Since \( y_j^* \geq 0 \), we have \( 0 \leq y_j^* < [x_j - d_j]^+ \) whenever the condition in the first case above holds. Therefore, we can replace \( [x_j - d_j]^+ \) in the first case by \( x_j - d_j \). On the other hand, the condition in the second case is equivalent to \( x_j - d_j \leq y_j^* \leq x_j \) and \( 0 \leq y_j^* \leq x_j \). Since \( y_j^* \geq 0 \), we can replace the condition in the second case by \( x_j - d_j \leq y_j^* \leq x_j \). These imply that we can write (8) as

\[
v_j(x_j, d_j) = \begin{cases} r_j d_j + \mathbb{E}\{ v_{j+1}(x_j - d_j, D_{j+1}) \} & \text{if } y_j^* < x_j - d_j \\ r_j [x_j - y_j^*] + \mathbb{E}\{ v_{j+1}(y_j^*, D_{j+1}) \} & \text{if } x_j - d_j \leq y_j^* \leq x_j \\ \mathbb{E}\{ v_{j+1}(x_j, D_{j+1}) \} & \text{if } x_j < y_j^* \end{cases}
\]

(9)

Therefore, it is easy to see that we can compute a stochastic subgradient of \( \mathbb{E}\{ v_j(\cdot, D_j) \} \) at \( x_j \) through the recursion

\[
\dot{v}_j(x_j, d_j) = \begin{cases} \mathbb{E}\{ \dot{v}_{j+1}(x_j - d_j, D_{j+1}) \} & \text{if } y_j^* < x_j - d_j \\ r_j & \text{if } x_j - d_j \leq y_j^* \leq x_j \\ \mathbb{E}\{ \dot{v}_{j+1}(x_j, D_{j+1}) \} & \text{if } x_j < y_j^* \end{cases}
\]

(10)

In the appendix, we formally show that (10) indeed gives a stochastic subgradient of \( \mathbb{E}\{ v_j(\cdot, D_j) \} \).

To construct tractable approximations to the stochastic subgradients of \( \{ f_j(\cdot) : j = 1, \ldots, n \} \), we mimic the computation in (10) by using the estimates of the optimal protection levels. In particular, letting \( \{ y_j^k : j = 1, \ldots, n \} \) be the estimates of the optimal protection levels at iteration \( k \) and using \( O(\cdot) \) to denote the operator that rounds a scalar to a nearest integer (by breaking ties in an arbitrary but consistent manner), we recursively define

\[
\rho_j^k(x_j, d_j, d_{j+1}, \ldots, d_n) = \begin{cases} \rho_{j+1}^k(x_j - d_j, d_{j+1}, \ldots, d_n) & \text{if } O(y_j^k) < x_j - d_j \\ r_j & \text{if } x_j - d_j \leq O(y_j^k) \leq x_j \\ \rho_{j+1}^k(x_j, d_{j+1}, \ldots, d_n) & \text{if } x_j < O(y_j^k) \
\end{cases}
\]

(11)
with \( \rho_{n+1}^k(\cdot) = 0 \). We propose using \( \rho_j^k(x_j, d_j, d_{j+1}, \ldots, d_n) \) to approximate \( \hat{v}_j(x_j, d_j) \). More specifically, at iteration \( k \), we replace \( \hat{v}_{j+1}(y_j, d_{j+1}) \) in (6) with \( \rho_{j+1}^k(y_j, d_{j+1}, \ldots, d_n) \) and use

\[
s_j^k(y_j, d_{j+1}, \ldots, d_n) = -r_j + \rho_{j+1}^k(y_j, d_{j+1}, \ldots, d_n)
\]

to approximate a stochastic subgradient of \( f_j(\cdot) \) at \( y_j \). Therefore, we propose the following algorithm to compute the optimal protection levels.

**Algorithm 1**

**Step 1.** Initialize the estimates of the optimal protection levels \( \{y_j^1 : j = 1, \ldots, n\} \) such that \( c \geq y_1^1 \geq y_2^1 \geq \ldots \geq y_n^1 = 0 \). Initialize the iteration counter by setting \( k = 1 \).

**Step 2.** Letting \( \{D_j^k : j = 1, \ldots, n\} \) be the demand random variables at iteration \( k \), set

\[
y_j^{k+1} = \max \left\{ \min \left\{ \left[ y_j^k + r_j s_j^k(y_j^k, D_{j+1}^k, \ldots, D_n^k) \right], c \right\}, O(y_j^{k+1}) \right\}
\]

for all \( j = 1, \ldots, n \), with \( y_{n+1}^{k+1} = 0 \).

**Step 3.** Increase \( k \) by 1 and go to Step 2.

We let \( \mathcal{F}_k \) be the filtration generated by \( \{\{y_1^1, \ldots, y_n^1\}, \{D_1^1, \ldots, D_n^1\}, \ldots, \{D_1^{k-1}, \ldots, D_n^{k-1}\}\} \). Given \( \mathcal{F}_k \), we assume that the conditional distribution of \( \{D_j^k : j = 1, \ldots, n\} \) is the same as the distribution of \( \{D_j : j = 1, \ldots, n\} \). We assume that the step size parameters \( \{\alpha_j^k : j = 1, \ldots, n\} \) are positive and \( \mathcal{F}_k \)-measurable, in which case the estimates of the optimal protection levels \( \{y_j^k : j = 1, \ldots, n\} \) are also \( \mathcal{F}_k \)-measurable. In the next section, we show that if the sequence \( \{y_j^k : j = 1, \ldots, n\}_k \) is generated by Algorithm 1, then it converges to a set of optimal protection levels w.p.1.

Several remarks are in order for our approximation to \( \hat{v}_j(x_j, d_j) \) and Algorithm 1. First, we need the realizations of the demand random variables \( \{D_j, D_{j+1}, \ldots, D_n\} \) to compute \( \rho_j^k(x_j, d_j, d_{j+1}, \ldots, d_n) \), whereas we only need the realization of the demand random variable \( D_j \) to compute \( \hat{v}_j(x_j, d_j) \). Also, the computation of \( \rho_j^k(x_j, d_j, d_{j+1}, \ldots, d_n) \) does not require computing expectations. Second, comparing (6) and (12) indicates that if \( \hat{v}_j(\cdot) \) and \( \mathbb{E}\{\rho_j^k(\cdot, \ldots, D_{j+1}^k, \ldots, D_n^k) \mid \mathcal{F}_k\} \) are close to each other for all \( j = 1, \ldots, n \), then the expected step directions \( \mathbb{E}\{\Delta_j(\cdot, D_{j+1}^k) \mid \mathcal{F}_k\} \) and \( \mathbb{E}\{s_j^k(\cdot, D_{j+1}^k, \ldots, D_n^k) \mid \mathcal{F}_k\} \) are close to each other. In this case, using the step direction \( s_j^k(y_j^k, D_{j+1}^k, \ldots, D_n^k) \) instead of \( \Delta_j(y_j^k, D_{j+1}^k) \) does not bring too much error in expectation. Our convergence proof is heavily based on this observation, which we make mathematically precise in Lemma 2 below. Third, we round the estimates of the optimal protection levels to nearest integers when computing \( \rho_j^k(x_j, d_j, d_{j+1}, \ldots, d_n) \). This becomes useful when we use Algorithm 1 in a real time setting where we use the protection levels \( \{O(y_j^k) : j = 1, \ldots, n\} \) to satisfy the demands from the fare classes and we update our estimates of the optimal protection levels after observing the demands. In such situations, it is important to use integer protection levels since we cannot sell a fraction of a seat. Fourth, the way that we update our estimates of the optimal protection levels in (13) ensures that we have \( c \geq O(y_1^k) \geq O(y_2^k) \geq \ldots \geq O(y_n^k) \geq 0 \) for all \( k = 1, 2, \ldots \). Therefore, the estimates of the optimal protection levels at each iteration are nested. It is also important to note that the update in (13) is of Gauss-Seidel variant. More specifically, we need the value of \( y_{j+1}^{k+1} \) to compute the value of \( y_j^{k+1} \). Therefore, Step 2 in Algorithm 1 has to be carried out starting from fare class \( n \) and moving backwards through the fare classes.
The step directions in Algorithm 1 are motivated by the dynamic programming formulation of the problem. On the other hand, the step directions in the method proposed by van Ryzin and McGill (2000) are motivated by the optimality conditions mentioned in the introduction. As we show in Lemma 2 below, our step directions are related to the stochastic subgradients of the value functions, but this is not the case for the step directions used by van Ryzin and McGill (2000). We also note that we work with discrete demand distributions. To deal with discrete demand distributions, van Ryzin and McGill (2000) propose a randomized version of their method, which randomly chooses between the protection levels $\mathcal{O}(y_j^k)$ and $\mathcal{O}(y_j^k) + 1$ at each iteration. The iterates of the randomized version converge to a set of optimal protection levels w.p.1, but the randomization between the protection levels $\mathcal{O}(y_j^k)$ and $\mathcal{O}(y_j^k) + 1$ results in suboptimality for the performance of the policy; see Section 4.2 in van Ryzin and McGill (2000). Finally, we emphasize that the estimates of the optimal protection levels that we obtain at each iteration are nested, whereas such a condition is not imposed by van Ryzin and McGill (2000).

5 Convergence Proof

In this section, we show that the iterates of Algorithm 1 converge to a set of optimal protection levels w.p.1. We begin with some preliminary results in Section 5.1 and complete the proof in Section 5.2.

5.1 Preliminaries

The next lemma establishes a uniform bound on our step directions.

Lemma 1 There exists a finite scalar $M$ such that we have

$$|\rho_j^k(x_j, D^k_j, D^k_{j+1}, \ldots, D^k_n)| \leq M \quad \text{and} \quad |s_j^k(x_j, D^k_{j+1}, \ldots, D^k_n)| \leq M$$

w.p.1 for all $x_j \in [0, c]$, $j = 1, \ldots, n$, $k = 1, 2, \ldots$.

Proof If we let $R = \max_{j \in \{1, \ldots, n\}} r_j$, then by using (11) and moving backwards through the fare classes, it is easy to see that $|\rho_j^k(x_j, D^k_j, D^k_{j+1}, \ldots, D^k_n)| \leq R$. By (12), the result follows by letting $M = 2R$. □

The next lemma shows that if the sequence $\{y_j^k : j = 1, \ldots, n\}$ get close to the optimal protection levels, then the step directions in Algorithm 1 are related to the stochastic subgradients of the value functions. In Lemma 2 and throughout the rest of the paper, since $\{y_j^k : j = 1, \ldots, n\}$ are $\mathcal{F}^k$-measurable, we treat $\{y_j^k : j = 1, \ldots, n\}$ as known constants when dealing with a conditional expectation of the form $\mathbb{E}\{\cdot | \mathcal{F}^k\}$. Also, since $\{v_j(\cdot, D_j) : j = 1, \ldots, n\}$ are piecewise-linear concave functions with points of nondifferentiability being a subset of integers, it is easy to see that $\mathcal{Y}_j^*$ in (5) is a closed interval with integer end points. We let $\mathcal{Y}_j^* = [L_j^*, U_j^*]$ throughout the rest of the paper, where $L_j^*$ and $U_j^*$ are integers.

Lemma 2 Assume that the sequence $\{y_j^k : j = 1, \ldots, n\}_k$ is generated by Algorithm 1. If it holds that

$$y_j^k \in (L_j^* - 1/2, U_j^* + 1/2), \quad y_{j+1}^k \in (L_{j+1}^* - 1/2, U_{j+1}^* + 1/2), \ldots, \quad y_n^k \in (L_n^* - 1/2, U_n^* + 1/2),$$

then we have $\mathbb{E}\{\rho_j^k(x_j, D^k_j, D^k_{j+1}, \ldots, D^k_n) | \mathcal{F}^k, D_j^k\} \in \partial v_j(x_j, D_j^k)$ w.p.1 for all $x_j \in [0, c]$. 8
Proof We show the result by induction over the fare classes. It is easy to show the result for fare class \( n \). Assuming that the result holds for fare class \( j + 1 \), we now show that the result holds for fare class \( j \). The assumption in the lemma implies that we can find \( y^*_j \in \mathcal{Y}^*_j \), \( y^*_{j+1} \in \mathcal{Y}^*_{j+1} \), ..., \( y^n \in \mathcal{Y}^*_n \) such that \( y^*_j = \mathcal{O}(y^*_j), y^*_{j+1} = \mathcal{O}(y^*_{j+1}), \ldots, y^n = \mathcal{O}(y^n) \). Taking the conditional expectations in \((11)\) and recalling that we use \( \hat{v}_{j+1}(x_j, d_{j+1}) \) to denote an element of \( \partial v_{j+1}(x_j, d_{j+1}) \), we obtain

\[
\mathbb{E}\{\rho_j^k(x_j, d_j, D_j^k, \ldots, D_n^k) \mid \mathcal{F}_j\} = \mathbb{E}\{\mathbb{E}\{\rho_j^k(x_j, d_j, D_j^k, \ldots, D_n^k) \mid \mathcal{F}_j, D_{j+1}^k\} \mid \mathcal{F}_j\}
\]

Comparing \((14)\) with \((10)\) and noting that the distribution of \( L^*_j \) is the same as the distribution of \( D_{j+1}^k \) conditional on \( \mathcal{F}_j \), we obtain \( \mathbb{E}\{\rho_j^k(x_j, d_j, D_j^k, \ldots, D_n^k) \mid \mathcal{F}_j\} \in \partial v_j(x_j, d_j) \).  

Roughly speaking, the next lemma shows that if the estimates of the optimal protection levels at iteration \( k \) are close to the optimal protection levels, then the estimates of the optimal protection levels at iteration \( k + 1 \) are also close to the optimal protection levels.

Lemma 3 Assume that the sequence \( \{y^k_j : j = 1, \ldots, n\}\) is generated by Algorithm 1. If it holds that

\[
y^k_j \in (L^*_j - 1/4, U^*_j + 1/4), y^k_{j+1} \in (L^*_{j+1} - 1/4, U^*_n + 1/4), \ldots, y^k_n \in (L^*_n - 1/4, U^*_n + 1/4) \text{ and } \alpha^k_j \in [0, 1/(4M)], \alpha^k_{j+1} \in [0, 1/(4M)], \ldots, \alpha^k_n \in [0, 1/(4M)],
\]

then we have \( \mathcal{O}(y^k_{j+1}) \in \mathcal{Y}^*_{j+1} \) w.p.1.

Proof All statements in the proof are in w.p.1 sense. We show the result by induction over the fare classes. Since \( r_n > 0 \), we have \( \mathcal{Y}^*_n = \{0\} \) by \((5)\) and \( s^k_n(\cdot) < 0 \) by \((12)\). Therefore, we have \( y^n = 0 \) by \((13)\) for all \( k = 1, 2, \ldots \) and the result holds for fare class \( n \). Assuming that the result holds for fare class \( j + 1 \), we now show that the result holds for fare class \( j \). By the assumption in the lemma and Lemma 1, we have

\[
L^*_j - 1/2 < y^*_{j+1} + \alpha^k_j s^k_j(y^k_j, D^k_{j+1}, \ldots, D^k_n) < U^*_j + 1/2.
\]

We consider three cases.

Case 1. Assume that \( y^k_j + \alpha^k_j s^k_j(y^k_j, D^k_{j+1}, \ldots, D^k_n) \geq c = \mathcal{O}(y^k_{j+1}) \). Since we have \( \mathcal{O}(y^k_{j+1}) \in \mathcal{Y}^*_{j+1} \subset [0, c] \) by the induction assumption, we obtain \( y^k_{j+1} = c \) by \((13)\). On the other hand, we have \( U^*_j + 1/2 > c \) by \((15)\). Since \( U^*_j \) is an integer smaller than \( c \), we obtain \( U^*_j = c \). Therefore, we have \( \mathcal{O}(y^k_{j+1}) = c \in [c, c] \subset [L^*_j, U^*_j] \).
We have the following convergence result for Algorithm 1. Assume that 

\[ y_j^k + \alpha_j^k s_j^k(y_j^k, D_{j+1}^k, \ldots, D_n^k) \geq O(y_{j+1}^{k+1}). \]

Since Algorithm 1 ensures that 

\[ y_j^k \geq 0 \quad \text{for all} \ j = 1, \ldots, n, \ k = 1, 2, \ldots, \] we have \( O(y_{j+1}^{k+1}) \geq 0. \) Therefore, by (13), we have \( y_j^{k+1} = y_j^k + \alpha_j^k s_j^k(y_j^k, D_{j+1}^k, \ldots, D_n^k) \). We obtain \( O(y_{j+1}^{k+1}) \in [L_j^*, U_j^*] \) by (15).

**Case 2.** Assume that 

\[ c > y_j^k + \alpha_j^k s_j^k(y_j^k, D_{j+1}^k, \ldots, D_n^k) \geq O(y_{j+1}^{k+1}). \]

\[ y_j^2 \geq \frac{c}{(4M)} \]

Case 2. Assume that 

\[ c \geq O(y_{j+1}^{k+1}) > y_j^k + \alpha_j^k s_j^k(y_j^k, D_{j+1}^k, \ldots, D_n^k). \]

We have \( y_j^{k+1} = O(y_{j+1}^{k+1}) \) by (13) and \( O(y_{j+1}^{k+1}) > L_j^* - 1/2 \) by (15). Since we have \( O(y_{j+1}^{k+1}) \in Y_{j+1}^* \) by the induction assumption, the fact that the optimal protection levels are nested implies that \( O(y_{j+1}^{k+1}) \leq U_j^* \). Therefore, we obtain \( L_j^* - 1/2 < y_j^{k+1} = O(y_{j+1}^{k+1}) \leq U_j^* \), which implies that \( O(y_{j+1}^{k+1}) \in [L_j^*, U_j^*] \).

In Section 5.2, we give a convergence result for Algorithm 1 that shows that the distance between \( y_j^k \) and the optimal protection level that is closest to \( y_j^k \) converges to zero w.p.1. For fare class \( j \), we define the optimal protection level that is closest to \( y_j^k \) as

\[ C_j(\hat{y}_j^k) = \arg\min_{y_j^* \in Y_j^*} |y_j^* - y_j^k|. \]

The next lemma shows a contraction type of result for Algorithm 1.

**Lemma 4** Assume that the sequence \{\( y_j^k : j = 1, \ldots, n \)\} is generated by Algorithm 1. If it holds that \( y_j^k \in (L_{j+1}^* - 1/4, U_{j+1}^* + 1/4) \), \( y_j^k + \alpha_j^k s_j^k(y_j^k, D_{j+1}^k, \ldots, D_n^k) \geq c \geq O(y_{j+1}^{k+1}) \). By the same argument in Lemma 3, we have \( y_j^{k+1} = c \). Since \( C_j(y_{j+1}^{k+1}) \) is the closest optimal protection level to \( y_j^{k+1} \) and \( U_j^* \leq c \), we obtain \( C_j(y_{j+1}^{k+1}) = U_j^* \). Using the fact that \( C_j(y_j^k) \leq U_j^* \leq c \), we have \( y_j^k + \alpha_j^k s_j^k(y_j^k, D_{j+1}^k, \ldots, D_n^k) \geq c = y_j^{k+1} \geq C_j(y_{j+1}^{k+1}) = U_j^* \) and the result follows.

**Proof** All statements in the proof are in w.p.1 sense. We consider the same three cases in the proof of Lemma 3.

**Case 1.** Assume that \( y_j^k + \alpha_j^k s_j^k(y_j^k, D_{j+1}^k, \ldots, D_n^k) \geq c \geq O(y_{j+1}^{k+1}) \). By the same argument in Lemma 3, we have \( y_j^{k+1} = c \). Since \( C_j(y_{j+1}^{k+1}) \) is the closest optimal protection level to \( y_j^{k+1} \) and \( U_j^* \leq c \), we obtain \( C_j(y_{j+1}^{k+1}) = U_j^* \). Using the fact that \( C_j(y_j^k) \leq U_j^* \leq c \), we have \( y_j^k + \alpha_j^k s_j^k(y_j^k, D_{j+1}^k, \ldots, D_n^k) \geq c = y_j^{k+1} \geq C_j(y_{j+1}^{k+1}) = U_j^* \) and the result follows.

**Case 2.** Assume that \( c > y_j^k + \alpha_j^k s_j^k(y_j^k, D_{j+1}^k, \ldots, D_n^k) \geq O(y_{j+1}^{k+1}) \). By the same argument in Lemma 3, we have \( y_j^{k+1} = y_j^k + \alpha_j^k s_j^k(y_j^k, D_{j+1}^k, \ldots, D_n^k) \). Therefore, we have \( |y_j^{k+1} - C_j(y_j^k)| = |y_j^{k+1} - C_j(y_j^k)| \), where the last inequality follows by (16).

**Case 3.** Assume that \( c \geq O(y_{j+1}^{k+1}) > y_j^k + \alpha_j^k s_j^k(y_j^k, D_{j+1}^k, \ldots, D_n^k) \). By the same argument in Lemma 3, we have \( y_j^{k+1} = O(y_{j+1}^{k+1}) \). If \( O(y_{j+1}^{k+1}) \in Y_j^* \), then \( |y_j^{k+1} - C_j(y_j^k)| = 0 \) and the result follows. We now assume that either \( O(y_{j+1}^{k+1}) > U_j^* \) or \( O(y_{j+1}^{k+1}) < L_j^* \). We immediately eliminate the former possibility, since we have \( O(y_{j+1}^{k+1}) \in Y_j^* \) by the assumption in the lemma and Lemma 3, which, together with the fact that the optimal protection levels are nested, implies that \( O(y_{j+1}^{k+1}) \leq U_j^* \). Therefore, we have \( y_j^k + \alpha_j^k s_j^k(y_j^k, D_{j+1}^k, \ldots, D_n^k) < O(y_{j+1}^{k+1}) = y_j^{k+1} < L_j^* \) and the result follows.

### 5.2 Convergence of Algorithm 1

We have the following convergence result for Algorithm 1.
Proposition 5 Assume that the sequence \{y^k_j : j = 1, \ldots, n\}_k is generated by Algorithm 1. If the sequence of step size parameters \{\alpha^k_j : j = 1, \ldots, n\}_k is positive and satisfies \(\sum_{k=1}^{\infty} \alpha^k_j = \infty\) and \(\sum_{k=1}^{\infty} |\alpha^k_j|^2 < \infty\) w.p.1 for all \(j = 1, \ldots, n\), then we have \(\lim_{k \to \infty} |y^k_j - C_j(y^k_j)| = 0\) w.p.1 for all \(j = 1, \ldots, n\).

Proof All statements in the proof are in w.p.1 sense. We show the result by induction over the fare classes. Since we have \(\mathcal{V}_n^* = \{0\}\) and \(y^k_k = 0\) for all \(k = 1, 2, \ldots\) by the argument in the proof of Lemma 3, the result holds for fare class \(n\). Assuming that the result holds for fare classes \(j + 1, j + 2, \ldots, n\), we now show that the result holds for fare class \(j\). The proof is in three parts. The first part shows that an inequality of the form \(\mathbb{E}\{Y^{k+1} | \mathcal{F}^k\} \leq Y^{k} - X^{k} + Z^{k}\) holds for appropriately defined sequences \(\{X^{k}\}_k, \{Y^{k}\}_k\) and \(\{Z^{k}\}_k\). The second part shows that \(\{X^{k}\}_k, \{Y^{k}\}_k\) and \(\{Z^{k}\}_k\) are positive and \(\mathcal{F}^k\)-measurable, and \(\{Z^{k}\}_k\) satisfies \(\sum_{k=1}^{\infty} Z^{k} < \infty\). In this case, we can conclude by the supermartingale convergence theorem that the sequence \(\{Y^{k}\}_k\) converges and we have \(\sum_{k=1}^{\infty} X^{k} < \infty\); see Neveu (1975). The third part uses these results to complete the proof.

Part 1. To capture the cases where the assumption of Lemma 4 holds, we define the event \(A^k_j\) as

\[
A^k_j = \{ y^k_{j+1} \in (L^*_{j+1} - 1/4, U^*_j + 1/4), y^k_{j+2} \in (L^*_{j+2} - 1/4, U^*_j + 1/4), \ldots, y^k_n \in (L^*_{n} - 1/4, U^*_n + 1/4) \}
\]

and \(\alpha^k_{j+1} \in [0, 1/(4M)], \alpha^k_{j+2} \in [0, 1/(4M)], \ldots, \alpha^k_n \in [0, 1/(4M)]\).

Using \(1(\cdot)\) to denote the indicator function, Lemma 4 and the fact that \(|y^k_{j+1} - C_j(y^k_{j+1})| \leq c\) imply that

\[
|y^k_{j+1} - C_j(y^k_{j+1})|^2 \\
\leq 1(A^k_j) |y^k_j + \alpha^k_j s^k_j(y^k_j, D^k_j, \ldots, D^k_n) - C_j(y^k_j)|^2 + [1 - 1(A^k_j)] c^2 \\
\leq |y^k_j - C_j(y^k_j)|^2 - 1(A^k_j) 2 \alpha^k_j [s^k_j(y^k_j, D^k_j, \ldots, D^k_n)]^2 [C_j(y^k_j) - y^k_j] + [\alpha^k_j]^2 M^2 + [1 - 1(A^k_j)] c^2,
\]

where the last inequality is by Lemma 1. Taking the conditional expectations and noting that \(1(A^k_j)\) is \(\mathcal{F}^k\)-measurable, we obtain

\[
\mathbb{E}\{ |y^k_{j+1} - C_j(y^k_{j+1})|^2 | \mathcal{F}^k \} \leq 1(A^k_j) |y^k_j - C_j(y^k_j)|^2 \\
- 1(A^k_j) 2 \alpha^k_j [C_j(y^k_j) - y^k_j] \mathbb{E}\{ s^k_j(y^k_j, D^k_j, \ldots, D^k_n) | \mathcal{F}^k \} + [\alpha^k_j]^2 M^2 + [1 - 1(A^k_j)] c^2.
\]

If we let \(Y^k = |y^k_j - C_j(y^k_j)|^2\), \(X^k = 1(A^k_j) 2 \alpha^k_j [C_j(y^k_j) - y^k_j] \mathbb{E}\{ s^k_j(y^k_j, D^k_j, \ldots, D^k_n) | \mathcal{F}^k \}\) and \(Z^k = [\alpha^k_j]^2 M^2 + [1 - 1(A^k_j)] c^2\), then the inequality above is of the form \(\mathbb{E}\{Y^k+1 | \mathcal{F}^k\} \leq Y^k - X^k + Z^k\).

Part 2. Clearly, \(\{Y^k\}_k\) and \(\{Z^k\}_k\) are positive and \(\{X^k\}_k\), \(\{Y^k\}_k\) and \(\{Z^k\}_k\) are \(\mathcal{F}^k\)-measurable.

We now show that \(\{X^k\}_k\) is positive. If \(1(A^k_j) = 0\), then we have \(X^k = 0\). If, on the other hand, we have \(1(A^k_j) = 1\), then we obtain \(\mathbb{E}\{ \rho^k_{j+1}(y^k_j, D^k_{j+1}, D^k_n) | \mathcal{F}^k, D^k_{j+1} \} \in \partial v_{j+1}(y^k_j, D^k_{j+1})\) by Lemma 2 and the definition of the event \(A^k_j\). Therefore, by (4) and (12), if \(1(A^k_j) = 1\), then we have \(\mathbb{E}\{ s^k_j(y^k_j, D^k_{j+1}, \ldots, D^k_n) | \mathcal{F}^k \} = -r_j + \mathbb{E}\{ \rho^k_{j+1}(y^k_j, D^k_{j+1}, \ldots, D^k_n) | \mathcal{F}^k, D^k_{j+1} \} | \mathcal{F}^k \} = -r_j + \mathbb{E}\{ v_{j+1}(y^k_j, D^k_{j+1}) | \mathcal{F}^k \} \) and \(\{s^k_j(y^k_j, D^k_{j+1}, \ldots, D^k_n) | \mathcal{F}^k \}\) is a subgradient of \(f_j(\cdot)\) at \(y^k_j\). In this case, we have \(|C_j(y^k_j) - y^k_j| \mathbb{E}\{ s^k_j(y^k_j, D^k_{j+1}, \ldots, D^k_n) | \mathcal{F}^k \} \geq f_j(C_j(y^k_j)) - f_j(y^k_j) \geq 0\), where the last inequality follows from the fact that \(C_j(y^k_j) \in \mathcal{V}^*_j\) and (5). Therefore, \(\{X^k\}_k\) is positive.
We now show that \( \sum_{k=1}^{\infty} Z^k < \infty \). Noting the induction assumption that \( \lim_{k \to \infty} |y_{j+1}^k - C_j(y_{j+1}^k)| = 0 \), \( \lim_{k \to \infty} |y_{j+2}^k - C_j(y_{j+2}^k)| = 0, \ldots, \lim_{k \to \infty} |y_n^k - C_n(y_n^k)| = 0 \) and the fact that \( \lim_{k \to \infty} \alpha_j^k = 0 \) for all \( j = 1, \ldots, n \), there exists a finite iteration counter \( K \) such that \( 1(A^K_j) = 1 \) for all \( k = K, K+1, \ldots \). Therefore, we have \( \sum_{k=1}^{\infty} Z^k \leq \sum_{k=1}^{\infty} (\alpha_j^k)^2 M^2 + K c^2 < \infty \).

**Part 3.** By the supermartingale convergence theorem and Parts 1 and 2, we conclude that the sequence \( \{ |y_j^k - C_j(y_j^k)| \}_k \) converges and we have \( \sum_{k=1}^{\infty} X^k < \infty \). Noting the discussion in Part 2, we have \( 1(A^K_j) = 1 \) and \( X^k = 2 \alpha_j^k |C_j(y_j^k) - y_j^k| \).\( \{ s_j D_{j_1^{(k)}}, \ldots, D_{j_n^{(k)}} \} | \mathcal{F}_j \} \geq 2 \alpha_j^k |f_j(C_j(y_j^k)) - f_j(y_j^k)| \) for all \( k = K, K+1, \ldots \). Therefore, we have \( \sum_{k=1}^{\infty} \alpha_j^k |f_j(C_j(y_j^k)) - f_j(y_j^k)| \leq \sum_{k=1}^{\infty} X^k < \infty \), which, together with the fact that \( \sum_{k=1}^{\infty} \alpha_j^k = \infty \), implies that \( \lim_{k \to \infty} f_j(C_j(y_j^k)) = f_j(y_j^k) = 0 \). Consequently, there exists a subsequence \( \{ \tilde{y}_j^k \}_k \) of \( \{ y_j^k \}_k \) such that \( \lim_{k \to \infty} |f_j(C_j(y_j^k)) - f_j(\tilde{y}_j^k)| = 0 \). Since the sequence \( \{ \tilde{y}_j^k \}_k \) takes values in the bounded interval \( [0, c] \), we can take a further subsequence \( \{ \tilde{\tilde{y}}_j^k \}_k \) such that \( \lim_{k \to \infty} \tilde{\tilde{y}}_j^k = \tilde{y}_j \) for some \( \tilde{y}_j \in [0, c] \).

Noting the definition of \( C_j(\cdot) \) and letting \( F_j^* = \max_{0 \leq y_j \leq c} f_j(y_j) \), we clearly have \( f_j(C_j(y_j^k)) = F_j^* \) for all \( j = 1, 2, \ldots \). Therefore, by the fact that \( \lim_{k \to \infty} |f_j(C_j(y_j^k)) - f_j(\tilde{y}_j^k)| = 0 \), we have \( \lim_{k \to \infty} f_j(y_j^k) = F_j^* \). On the other hand, by the continuity of \( f_j(\cdot) \) and the fact that \( \lim_{k \to \infty} y_j^k = \tilde{y}_j \), we have \( \lim_{k \to \infty} f_j(\tilde{y}_j^k) = f_j(\tilde{y}_j) \). From the last two statements, we obtain \( f_j(\tilde{y}_j) = F_j^* \) and \( \tilde{y}_j \in \mathcal{Y}_j^* \), which imply that \( |\tilde{y}_j^k - C_j(y_j^k)| \leq |\tilde{y}_j^k - \tilde{y}_j| \) for all \( k = 1, 2, \ldots \). Therefore, since \( \{ |\tilde{y}_j^k - \tilde{y}_j| \}_k \) converges to zero, \( \{ |y_j^k - C_j(y_j^k)| \}_k \) also converges to zero. Recalling that the whole sequence \( \{ |y_j^k - C_j(y_j^k)| \}_k \) converges, we obtain \( \lim_{k \to \infty} |y_j^k - C_j(y_j^k)| = 0 \).

A simple corollary to Proposition 5 is that there exists a finite iteration number \( K \) w.p.1 such that we have \( L_j - 1/2 < y_j^k < U_j + 1/2 \) for all \( j = 1, \ldots, n, k = K, K+1, \ldots \). Therefore, we have \( \mathcal{O}(y_j^k) \in \mathcal{Y}_j^* \) for all \( j = 1, \ldots, n, k = K, K+1, \ldots \) and the policy that uses \( \{ \mathcal{O}(y_j^k) : j = 1, \ldots, n \} \) as the protection levels is optimal w.p.1 after a finite number of iterations. As mentioned before, the randomized version of the method proposed by van Ryzin and McGill (2000) does not guarantee that the performance of the policy is optimal in the limit, although the iterates of this method converge to a set of optimal protection levels w.p.1. To illustrate this on a simple example, Figure 2 plots the total expected revenues corresponding to the protection levels used by the method proposed by van Ryzin and McGill (2000) and Algorithm 1 as a function of the iteration counter. The performance of Algorithm 1 is eventually optimal, whereas...
the performance of the method proposed by van Ryzin and McGill (2000) fluctuates.

6 Censored Demands

Demand censorship refers to the situation where we can observe the number of seats sold to a fare class, but not the actual amount of demand from a fare class. In this case, our demand observations are truncated when the amount of demand from a fare class exceeds the number of seats that we make available for sale to a fare class. We begin this section with a negative result that shows that the step direction in (12) cannot be computed when the demand information is censored. This implies that Algorithm 1 becomes inapplicable under censored demands. We then propose two alternative versions of Algorithm 1 that remain applicable under censored demands. The first alternative version has a somewhat weak convergence guarantee. The second alternative version is a heuristic modification of the first one, but it has somewhat more desirable practical performance.

If the demand information is censored, then we do not observe the demand random variables \( \{D_j^k : j = 1, \ldots, n\} \) in Step 2 of Algorithm 1. Instead, we simulate the behavior of the policy characterized by the protection levels \( \{O(y^k_j) : j = 1, \ldots, n\} \) and observe the number of seats sold to different fare classes. In this case, Step 2 of Algorithm 1 has to be replaced with the following steps.

**Step 2.a.** Set the initial capacity \( x^k_1 \) to \( c \) and set \( j = 1 \).

**Step 2.b.** Make \( [x^k_1 - O(y^k_1)]^+ \) seats available for sale to fare class \( j \).

**Step 2.c.** Observe the number of seats sold to fare class \( j \) as \( \min\{[x^k_j - O(y^k_j)]^+, D^j_k\} \) and compute the capacity just before making the decisions for fare class \( j+1 \) as \( x_{j+1}^k = x^k_j - \min\{[x^k_j - O(y^k_j)]^+, D^j_k\} \).

**Step 2.d.** If \( j < n \), then increase \( j \) by 1 and go to Step 2.b.

**Step 2.e.** For all \( j = 1, \ldots, n \), set \( y^k_j = \max\{\min\{y^k_j + \alpha^k_s(y^k_j, D^k_{j-1}, \ldots, D^k_n), c\}, O(y^k_{j+1})\} \).

Therefore, we only have access to \( \{\min\{[x^k_j - O(y^k_j)]^+, D^k_j\} : j = 1, \ldots, n\} \), but not the demand random variables themselves. Unfortunately, this information is not adequate to compute \( s^k_j(y^k_1, D^k_2, \ldots, D^k_n) \) for all \( j = 1, \ldots, n \) and Algorithm 1 becomes inapplicable when the demand information is censored.

To illustrate, we consider a numerical example with \( n = 3 \), \( c = 4 \), \( y^1_1 = 3.2 \), \( y^2_2 = 2.1 \), \( y^3_3 = 0 \), \( D^1_1 = 1 \), \( D^1_2 = 1 \) and \( D^1_3 = 2 \). By Steps 2.a-2.d above, we have \( x^k_1 = 4 \), \( \min\{[x^k_1 - O(y^k_1)]^+, D^k_1\} = 1 \), \( x^k_2 = 4 - 1 = 3 \), \( \min\{[x^k_2 - O(y^k_2)]^+, D^k_2\} = 1 \), \( x^k_3 = 3 - 1 = 2 \) and \( \min\{[x^k_3 - O(y^k_3)]^+, D^k_3\} = 2 \). By (12), computing \( s^k(y^k_1, D^k_2, D^k_3) \) requires computing \( \rho^k_2(y^k_1, D^k_2, D^k_3) \) and we have

\[
\rho^k_2(y^k_1, D^k_2, D^k_3) = \begin{cases} 
\rho^k_3(y^k_1 - D^k_2, D^k_3) & \text{if } O(2.1) < 3.2 - D^k_2 \\
\frac{3}{2} & \text{if } 3.2 - D^k_2 \leq O(2.1) \leq 3.2
\end{cases}
\]

by (11). Therefore, to compute \( s^k(y^k_1, D^k_2, D^k_3) \), we need to know whether \( D^k_2 < 1.2 \) or \( D^k_2 \geq 1.2 \). However, if we only have access to \( \{y^k_1, y^k_2, y^k_3\} \), \( \{x^k_1, x^k_2, x^k_3\} \) and \( \min\{[x^k_1 - O(y^k_1)]^+, D^k_1\}, \min\{[x^k_2 - O(y^k_2)]^+, D^k_2\}, \min\{[x^k_3 - O(y^k_3)]^+, D^k_3\}\}, \) then we know that \( 1 = \min\{[x^k_2 - O(y^k_2)]^+, D^k_2\} = \min\{1, D^k_2\} \leq D^k_2 \), but not whether \( D^k_2 < 1.2 \) or \( D^k_2 \geq 1.2 \). In the next four sections, we describe different ways to deal with this difficulty.
6.1 Using Fractional Estimates of the Protection Levels

One obvious approach to deal with the censored demands is to stop rounding the estimates of the optimal protection levels. This amounts to dropping all $O(\cdot)$ operators throughout the paper. In this case, it is possible to modify Proposition 5 in an obvious manner to get a convergence guarantee and it is easy to check that having access to $\{y_j^k: j = 1, \ldots, n\}$, $\{x_j^k: j = 1, \ldots, n\}$ and $\{\min\{[x_j^k - O(y_j^k)]^+, D_j^k\} : j = 1, \ldots, n\}$ is adequate to compute $s_j^k(y_j^k, D_{j+1}^k, \ldots, D_n^k)$. However, this approach uses fractional protection levels in Steps 2.a-2.e and it is not useful when we use Algorithm 1 in a real time setting.

6.2 Using a Relaxed View of Demand Censorship

It is possible to deal with the censored demands under a somewhat relaxed view of demand censorship. This relaxed view of demand censorship assumes that we have access to $\{y_j^k: j = 1, \ldots, n\}$, $\{x_j^k: j = 1, \ldots, n\}$ and $\{\min\{[x_j^k - O(y_j^k)]^+ + 1, D_j^k\} : j = 1, \ldots, n\}$, which amounts to assuming that we can observe the number of seats sold if we were to make $[x_j^k - O(y_j^k)]^+ + 1$ seats available for sale to fare class $j$. In other words, the relaxed view of demand censorship assumes that we can observe whether an extra seat would have been sold to a fare class if it had been made available. Therefore, the relaxed view of demand censorship effectively assumes that we can observe whether the demand from a fare class strictly exceeds the number of seats that we make available for sale. The next proposition shows that we can compute $s_j^k(y_j^k, D_{j+1}^k, \ldots, D_n^k)$ under the relaxed view of demand censorship. This implies that Algorithm 1 remains applicable with no modifications under the relaxed view of demand censorship.

Proposition 6 Having access to $\{y_j^k: j = 1, \ldots, n\}$, $\{x_j^k: j = 1, \ldots, n\}$ and $\{\min\{[x_j^k - O(y_j^k)]^+ + 1, D_j^k\} : j = 1, \ldots, n\}$ is adequate to compute $s_j^k(y_j^k, D_{j+1}^k, \ldots, D_n^k)$ for all $j = 1, \ldots, n$.

Proof We use a constructive proof, which, we think, shows the computations involved more clearly. We first use induction over the fare classes to show that

$$x_j^k \geq O(y_{j-1}^k) \geq O(y_j^k)$$

(17)

for all $j = 2, \ldots, n$. This is easy to show for the second fare class. Assuming that the result holds for fare class $j$, we have $x_{j+1}^k = x_j^k - \min\{[x_j^k - O(y_j^k)]^+, D_j^k\} = x_j^k - \min\{x_j^k - O(y_j^k), D_j^k\} = \max\{O(y_j^k), x_j^k - D_j^k\} \geq O(y_j^k) \geq O(y_{j+1}^k)$, where the last inequality uses the fact that Algorithm 1 ensures that the estimates of the optimal protection levels are nested. We now focus on computing $s_j^k(y_j^k, D_{j+1}^k, \ldots, D_n^k)$. By (12), this requires computing $\rho_{j+1}^k(y_j^k, D_{j+1}^k, \ldots, D_n^k)$. We consider two cases.

Case 1. Assume that $\min\{[x_{j+1}^k - O(y_{j+1}^k)]^+ + 1, D_{j+1}^k\} = [x_{j+1}^k - O(y_{j+1}^k)]^+ + 1$. In this case, we deduce that $D_{j+1}^k \geq [x_{j+1}^k - O(y_{j+1}^k)]^+ + 1 \geq O(y_j^k) - O(y_{j+1}^k) + 1 \geq y_j^k - O(y_{j+1}^k)$, where the second inequality follows from (17). This chain of inequalities and (13) imply that $y_j^k - D_{j+1}^k \leq O(y_{j+1}^k) \leq y_j^k$ and we obtain $\rho_{j+1}^k(y_j^k, D_{j+1}^k, D_{j+2}^k, \ldots, D_n^k) = r_{j+1}$ by (11).

Case 2. Assume that $\min\{[x_{j+1}^k - O(y_{j+1}^k)]^+ + 1, D_{j+1}^k\} < [x_{j+1}^k - O(y_{j+1}^k)]^+ + 1$. In this case, we deduce that $D_{j+1}^k = \min\{[x_{j+1}^k - O(y_{j+1}^k)]^+ + 1, D_{j+1}^k\}$. Therefore, we have access to the value of $D_{j+1}^k$. We consider two subcases.
Case 2.a. Assume that $D_{j+1}^k \geq y_j^k - \mathcal{O}(y_j^{k+1})$. We have $y_j^k - D_{j+1}^k \leq \mathcal{O}(y_j^{k+1}) \leq y_j^k$, where the second inequality follows from the same argument in Case 1 and we obtain $\rho_{j+1}^k(y_j^k, D_{j+1}^k, D_{j+2}^k, \ldots, D_n^k) = r_{j+1}$.

Case 2.b. Assume that $D_{j+1}^k < y_j^k - \mathcal{O}(y_j^{k+1})$. We have $\rho_{j+1}^k(y_j^k, D_{j+1}^k, D_{j+2}^k, \ldots, D_n^k) = \rho_{j+2}^k(y_j^k - D_{j+1}^k, D_{j+2}^k, \ldots, D_n^k)$ by (11).

Therefore, if Cases 1 or 2.a holds, then we are done. Otherwise, it remains to compute $\rho_{j+2}^k(y_j^k - D_{j+1}^k, D_{j+2}^k, \ldots, D_n^k)$ for a known value of $D_{j+1}^k$. The result follows by continuing in the same fashion for the subsequent fare classes. For example, assume that Case 2.b holds, in which case it remains to compute $\rho_{j+2}^k(y_j^k - D_{j+1}^k, D_{j+2}^k, \ldots, D_n^k)$ for a known value of $D_{j+1}^k$. Since Case 2.b is a subcase of Case 2, we have $D_{j+1}^k = \min\{[x_{j+1}^k - \mathcal{O}(y_{j+1}^k)]^+, D_{j+1}^k\}$ and we obtain $x_{j+2}^k = x_{j+1}^k - \min\{[x_{j+1}^k - \mathcal{O}(y_{j+1}^k)]^+, D_{j+1}^k\} = x_{j+1}^k - D_{j+1}^k$. We consider two cases similar to Cases 1 and 2.

Case I. Assume that $\min\{[x_{j+1}^k - \mathcal{O}(y_{j+1}^k)]^+, D_{j+1}^k\} = [x_{j+2}^k - \mathcal{O}(y_{j+2}^k)]^+ + 1$. Using (17), we obtain $D_{j+2}^k \geq [x_{j+2}^k - \mathcal{O}(y_{j+2}^k)]^+ + 1 = x_{j+2}^k - \mathcal{O}(y_{j+2}^k) + 1 = x_{j+1}^k - D_{j+1}^k - \mathcal{O}(y_{j+2}^k) + 1 \geq \mathcal{O}(y_j^k) - D_{j+1}^k - \mathcal{O}(y_{j+2}^k) + 1 \geq y_j^k - D_{j+1}^k - \mathcal{O}(y_{j+2}^k)$. Since we assume that Case 2.b holds, (17) also implies that $D_{j+1}^k < y_j^k - \mathcal{O}(y_{j+1}^k) \leq y_j^k - \mathcal{O}(y_{j+2}^k)$. Therefore, we obtain $y_j^k - D_{j+1}^k - D_{j+2}^k \leq \mathcal{O}(y_{j+2}^k) \leq y_j^k - D_{j+1}^k$ and we have $\rho_{j+2}^k(y_j^k - D_{j+1}^k, D_{j+2}^k, D_{j+3}^k, \ldots, D_n^k) = r_{j+2}$ by (11).

Case II. Assume that $\min\{[x_{j+2}^k - \mathcal{O}(y_{j+2}^k)]^+, D_{j+2}^k\} < [x_{j+2}^k - \mathcal{O}(y_{j+2}^k)]^+ + 1$. In this case, we deduce that $D_{j+2}^k = \min\{[x_{j+2}^k - \mathcal{O}(y_{j+2}^k)]^+, D_{j+2}^k\}$ and we have access to the value of $D_{j+2}^k$. Since Case 2.b is a subcase of Case 2, we also have access to the value of $D_{j+1}^k$. We consider two subcases similar to Cases 2.a and 2.b.

Case II.a. Assume that $D_{j+1}^k + D_{j+2}^k \geq y_j^k - \mathcal{O}(y_{j+2}^k)$. We have $y_j^k - D_{j+1}^k - D_{j+2}^k \leq \mathcal{O}(y_{j+2}^k) \leq y_j^k - D_{j+1}^k$, where the second inequality follows from the same argument in Case I and we obtain $\rho_{j+2}^k(y_j^k - D_{j+1}^k, D_{j+2}^k, D_{j+3}^k, \ldots, D_n^k) = r_{j+2}$ by (11).

Case II.b. Assume that $D_{j+1}^k + D_{j+2}^k < y_j^k - \mathcal{O}(y_{j+2}^k)$. We have $\rho_{j+2}^k(y_j^k - D_{j+1}^k, D_{j+2}^k, D_{j+3}^k, \ldots, D_n^k) = \rho_{j+3}^k(y_j^k - D_{j+1}^k - D_{j+2}^k, D_{j+3}^k, \ldots, D_n^k)$ by (11).

Therefore, if Cases I or II.a holds, then we are done. Otherwise, it remains to compute $\rho_{j+3}^k(y_j^k - D_{j+1}^k - D_{j+2}^k, D_{j+3}^k, \ldots, D_n^k)$ for known values of $D_{j+1}^k$ and $D_{j+2}^k$. As mentioned before, the result follows by continuing in the same fashion for the subsequent fare classes.

Therefore, Algorithm 1 is applicable under the relaxed view of demand censorship.

6.3 Perturbing the Demand Random Variables

In certain practical settings, it may not be possible to adopt the relaxed view of demand censorship described in the previous section and Algorithm 1 becomes inapplicable. In this section, we develop an alternative version of Algorithm 1 that is applicable under the assumption that we only have access to $\{y_j^k : j = 1, \ldots, n\}$, $\{x_j^k : j = 1, \ldots, n\}$ and $\min\{[x_j^k - \mathcal{O}(y_j^k)]^+, D_j^k\} : j = 1, \ldots, n\}. Therefore, the alternative version is applicable without the relaxed view of demand censorship. The alternative version, however, does not converge to the optimum protection levels, but we show that the distance
between its iterates and the optimal protection levels is bounded by \( n \) in the limit w.p.1.

The alternative version of Algorithm 1 that we propose in this section is obtained by replacing Step 2 of Algorithm 1 with the following steps.

**Algorithm 2**

**Step 2.a.** Set the initial capacity \( x^k_j \) to \( c \) and set \( j = 1 \).

**Step 2.b.** Make \( [x^k_j + O(y_j^k)]^+ \) seats available for sale to fare class \( j \).

**Step 2.c.** Observe the number of seats sold to fare class \( j \) as \( \min\{[x^k_j - O(y_j^k)]^+, D^k_j\} \) and compute the capacity just before making the decisions for fare class \( j + 1 \) as \( x^k_{j+1} = x^k_j - \min\{[x^k_j - O(y_j^k)]^+, D^k_j\} \).

**Step 2.d.** If \( j < n \), then increase \( j \) by 1 and go to Step 2.b.

**Step 2.e.** For all \( j = 1, \ldots, n \), set

\[
y_j^{k+1} = \max\{\min\{[y_j^k + \alpha^k_j s_j^k(y_j^k, D_{j+1}^k + 1, \ldots, D_n^k + 1)]^+, c\}, O(y_j^{k+1})\}. \tag{18}
\]

In Steps 2.a-2.d, Algorithm 2 uses the demand random variables \( \{D_j^k : j = 1, \ldots, n\} \) to simulate the behavior of the policy characterized by the protection levels \( \{O(y_j^k) : j = 1, \ldots, n\} \). In Step 2.e, however, it uses the demand random variables \( \{D_j^k + 1 : j = 1, \ldots, n\} \) to compute the step direction. Therefore, Algorithm 2 uses incorrect demand random variables when updating the estimates of the optimal protection levels. For this reason, the iterates of Algorithm 2 do not necessarily converge to the optimal protection levels. Nevertheless, as the next corollary to Proposition 6 shows, having access to \( \{y_j^k : j = 1, \ldots, n\} \), \( \{x_j^k : j = 1, \ldots, n\} \) and \( \{\min\{[x_j^k - O(y_j^k)]^+, D_j^k\} : j = 1, \ldots, n\} \) is adequate to compute \( s_j^k(y_j^k, D_{j+1}^k + 1, \ldots, D_n^k + 1) \). Therefore, Algorithm 2 is applicable when the demand information is censored and we cannot adopt the relaxed view of demand censorship.

**Corollary 7** Having access to \( \{y_j^k : j = 1, \ldots, n\}, \{x_j^k : j = 1, \ldots, n\} \) and \( \{\min\{[x_j^k - O(y_j^k)]^+, D_j^k\} : j = 1, \ldots, n\} \) is adequate to compute \( s_j^k(y_j^k, D_{j+1}^k + 1, \ldots, D_n^k + 1) \) for all \( j = 1, \ldots, n \).

**Proof** Replacing \( \{D_j^k : j = 1, \ldots, n\} \) in Proposition 6 by \( \{D_j^k + 1 : j = 1, \ldots, n\} \), we know that having access to \( \{y_j^k : j = 1, \ldots, n\}, \{x_j^k : j = 1, \ldots, n\} \) and \( \{\min\{[x_j^k - O(y_j^k)]^+, D_j^k + 1\} : j = 1, \ldots, n\} \) is adequate to compute \( s_j^k(y_j^k, D_{j+1}^k + 1, \ldots, D_n^k + 1) \). Clearly, if we know the values of \( \{\min\{[x_j^k - O(y_j^k)]^+, D_j^k\} : j = 1, \ldots, n\} \), then we know the values of \( \{\min\{[x_j^k - O(y_j^k)]^+ + 1, D_j^k + 1\} : j = 1, \ldots, n\} \). Therefore, having access to \( \{y_j^k : j = 1, \ldots, n\}, \{x_j^k : j = 1, \ldots, n\} \) and \( \{\min\{[x_j^k - O(y_j^k)]^+, D_j^k\} : j = 1, \ldots, n\} \) is adequate to compute \( s_j^k(y_j^k, D_{j+1}^k + 1, \ldots, D_n^k + 1) \).

The next corollary to Proposition 5 gives a somewhat weak convergence result for Algorithm 2.

**Corollary 8** Assume that the sequence \( \{y_j^k : j = 1, \ldots, n\}_k \) is generated by Algorithm 2. If the sequence of step size parameters \( \{\alpha_j^k : j = 1, \ldots, n\}_k \) is positive and satisfies \( \sum_{k=1}^{\infty} \alpha_j^k = \infty \) and \( \sum_{k=1}^{\infty} \alpha_j^k \) < \( \infty \) w.p.1 for all \( j = 1, \ldots, n \), then there exists a finite iteration number \( K \) w.p.1 such that we have \( O(y_j^k) \in [L_j^k, U_j^k + n - j] \) for all \( j = 1, \ldots, n, k = K, K + 1, \ldots \).

**Proof** We sketch the main ideas of the proof here and defer the details to the appendix. As far as updating the estimates of the optimal protection levels in Step 2.e is concerned, Algorithm 2 assumes
that the demand random variables are \( \{D^k_j + 1 : j = 1, \ldots, n\} \). Therefore, the iterates of Algorithm 2 converge to the optimal protection levels for the problem

\[
\tilde{v}_j(x_j, d_j) = \max_{0 \leq u_j \leq \min\{x_j, d_j + 1\}} r_j u_j + \mathbb{E}\{\tilde{v}_{j+1}(x_j - u_j, D_{j+1})\} \tag{19}
\]

in the sense of Proposition 5. If we use \( \hat{Y}^*_j \) to denote the set of optimal protection levels when making the decisions for fare class \( j \) in the problem above, then it is possible to show that \( \hat{Y}^*_j \subset [L^*_j, U^*_j + n - j] \) and the result follows.

The result in Corollary 8 can be weak especially when the number of fare classes is large. Furthermore, the bound on the protection levels does not imply a bound on the total expected revenue obtained by the corresponding policy. Nevertheless, Brumelle and McGill (1993) and Robinson (1995) provide some evidence that the total expected revenue is robust to small deviations from the optimal protection levels.

In the next section, we combine the ideas in Sections 6.2 and 6.3 to develop another alternative version of Algorithm 1. This alternative version is also applicable when the demand information is censored and has somewhat more desirable practical performance than Algorithm 2. However, it does not have a convergence guarantee.

### 6.4 Perturbing the Demand Random Variables When Necessary

The main motivation for using the demand random variables \( \{D^k_j + 1 : j = 1, \ldots, n\} \) in Algorithm 2 is that this allows us to compute the step direction in Step 2.e under censored demands. It turns out that we do not need to increase all demand random variables by one to be able to compute the step direction.

In this section, we propose an alternative version of Algorithm 1, which is obtained by replacing Step 2 of Algorithm 1 with the following steps.

**Algorithm 3**

**Step 2.a.** Set the initial capacity \( x^k_j \) to \( c \) and set \( j = 1 \).

**Step 2.b.** Make \( [x^k_j - \mathcal{O}(y^k_j)]^+ \) seats available for sale to fare class \( j \).

**Step 2.c.** Observe the number of seats sold to fare class \( j \) as \( \min\{[x^k_j - \mathcal{O}(y^k_j)]^+, D^k_j\} \) and compute the capacity just before making the decisions for fare class \( j + 1 \) as \( x^k_{j+1} = x^k_j - \min\{[x^k_j - \mathcal{O}(y^k_j)]^+, D^k_j\} \).

Also, set

\[
\tilde{d}^k_j = \begin{cases} 
D^k_j & \text{if } \min\{[x^k_j - \mathcal{O}(y^k_j)]^+, D^k_j\} < [x^k_j - \mathcal{O}(y^k_j)]^+ \\
D^k_j + 1 & \text{if } \min\{[x^k_j - \mathcal{O}(y^k_j)]^+, D^k_j\} = [x^k_j - \mathcal{O}(y^k_j)]^+. 
\end{cases} \tag{20}
\]

**Step 2.d.** If \( j < n \), then increase \( j \) by 1 and go to Step 2.b.

**Step 2.e.** For all \( j = 1, \ldots, n \), set

\[
y^{k+1}_j = \max\{\min\{[y^k_j + \alpha^k_j s^k_j(y^k_j, \tilde{d}^k_{j+1}, \ldots, \tilde{d}^k_n)^+], c\}, \mathcal{O}(y^{k+1}_{j+1})\}. \tag{21}
\]

We note that \( \tilde{d}^k_j \) is equal to \( D^k_j \) when the number of seats sold to fare class \( j \) is strictly less than the number of seats made available for sale to fare class \( j \) and it is equal to \( D^k_j + 1 \) when the number of seats sold to fare class \( j \) is equal to the number of seats made available for sale to fare class \( j \). Comparing (18)
and (21), since \( \tilde{d}_j^k \) is not always equal to \( D_j^k + 1 \), the hope is that the step direction used by Algorithm 3 is "closer" to \( s_j^k(y_j^k, D_{j+1}^k, \ldots, D_n^k) \) than the step direction used by Algorithm 2.

Contrary to what our description of Algorithm 3 suggests, it is, in fact, not necessary to compute \( \{\tilde{d}_j^k : j = 1, \ldots, n\} \) in Step 2.c explicitly. In particular, the next corollary to Proposition 6 shows that having access to \( \{y_j^k : j = 1, \ldots, n\} \), \( \{x_j^k : j = 1, \ldots, n\} \) and \( \{\min\{|x_j^k - \mathcal{O}(y_j^k)|, D_j^k\} : j = 1, \ldots, n\} \) is adequate to compute \( s_j^k(y_j^k, \tilde{d}_{j+1}^k, \ldots, \tilde{d}_n^k) \). Therefore, since \( \{\tilde{d}_j^k : j = 1, \ldots, n\} \) are only used in the computation of \( s_j^k(y_j^k, \tilde{d}_{j+1}^k, \ldots, \tilde{d}_n^k) \) in (21), it is not necessary to compute \( \{\tilde{d}_j^k : j = 1, \ldots, n\} \) explicitly. The proof of the next corollary is similar to that of Corollary 7 and is deferred to the appendix.

**Corollary 9** If we let \( \{\tilde{d}_j^k : j = 1, \ldots, n\} \) be as in (20), then having access to \( \{y_j^k : j = 1, \ldots, n\} \), \( \{x_j^k : j = 1, \ldots, n\} \) and \( \{\min\{|x_j^k - \mathcal{O}(y_j^k)|, D_j^k\} : j = 1, \ldots, n\} \) is adequate to compute \( s_j^k(y_j^k, \tilde{d}_{j+1}^k, \ldots, \tilde{d}_n^k) \) for all \( j = 1, \ldots, n \).

Therefore, Corollary 9 shows that Algorithm 3 is applicable when the demand information is censored.

## 7 Numerical Experiments

In this section, we compare the performances of Algorithms 1-3 with the performance of the stochastic approximation method proposed by van Ryzin and McGill (2000). Our test problems are taken from van Ryzin and McGill (2000) and we introduce some variety by using the approach followed by Huh and Rusmevichientong (2006). Specifically, we work with test problems that involve 4, 8 and 12 fare classes. The demand from each fare class is normally distributed, and to satisfy our assumptions, we discretize the demand random variables by rounding them to the nearest integer. Table 1 gives the revenues associated with the fare classes, along with the means and standard deviations of the demand random variables. For the test problem with 4 fare classes, we use \( c \in \{124, 164\} \), in which case the total expected demand is 25% more or 5% less than the available capacity. Similarly, for the test problems with 8 and 12 fare classes, we respectively use \( c \in \{260, 344\} \) and \( c \in \{409, 541\} \).

<table>
<thead>
<tr>
<th>fare class</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>revenue</td>
<td>1050</td>
<td>567</td>
<td>527</td>
<td>350</td>
</tr>
<tr>
<td>mean</td>
<td>17.3</td>
<td>45.1</td>
<td>73.6</td>
<td>19.8</td>
</tr>
<tr>
<td>std. dev</td>
<td>5.8</td>
<td>15.1</td>
<td>17.4</td>
<td>6.6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>fare class</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>revenue</td>
<td>1155</td>
<td>1050</td>
<td>623.7</td>
<td>579.7</td>
<td>567.7</td>
<td>527</td>
<td>385</td>
<td>350</td>
</tr>
<tr>
<td>mean</td>
<td>19</td>
<td>17.3</td>
<td>49.6</td>
<td>81</td>
<td>45.1</td>
<td>73.6</td>
<td>21.8</td>
<td>19.8</td>
</tr>
<tr>
<td>std. dev</td>
<td>6.4</td>
<td>5.8</td>
<td>16.5</td>
<td>81</td>
<td>15.7</td>
<td>17.4</td>
<td>7.3</td>
<td>6.6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>fare class</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>revenue</td>
<td>1260</td>
<td>1155</td>
<td>1050</td>
<td>680.4</td>
<td>632.4</td>
<td>623.7</td>
<td>579.7</td>
<td>567</td>
<td>420</td>
<td>385</td>
<td>350</td>
<td></td>
</tr>
<tr>
<td>mean</td>
<td>20.8</td>
<td>19</td>
<td>17.3</td>
<td>54.1</td>
<td>88.3</td>
<td>49.6</td>
<td>81</td>
<td>45.1</td>
<td>73.6</td>
<td>23.8</td>
<td>21.8</td>
<td>19.8</td>
</tr>
<tr>
<td>std. dev</td>
<td>7</td>
<td>6.4</td>
<td>5.8</td>
<td>18</td>
<td>20.9</td>
<td>16.5</td>
<td>19.1</td>
<td>15</td>
<td>17.4</td>
<td>7.9</td>
<td>7.3</td>
<td>6.6</td>
</tr>
</tbody>
</table>
For Algorithms 1-3 and for the method proposed by van Ryzin and McGill (2000), we test three strategies to choose the initial protection levels. In particular, we use

\[ y_j^1 = \frac{r_{j+1} + \ldots + r_n}{r_1 + \ldots + r_n} \cdot c \quad (22) \]

\[ y_j^1 = \frac{\mu_{j+1} + \ldots + \mu_n}{\mu_1 + \ldots + \mu_n} \cdot c \quad (23) \]

\[ y_j^1 = \frac{\mu_{j+1} r_{j+1} + \ldots + \mu_n r_n}{\mu_1 r_1 + \ldots + \mu_n r_n} \cdot c, \quad (24) \]

where \( \mu_j \) is the expected value of the demand from fare class \( j \). These initial protection levels are heuristically motivated by the observation that if the revenue associated with a fare class is large and the demand from a fare class is likely to be large, then we should protect a large number of seats for this fare class. We refer to the sets of initial protection levels computed by (22), (23) and (24) respectively as \( Y^1_R \), \( Y^1_M \) and \( Y^1_{MR} \). Although there is no guarantee, for our test problems, the total expected revenues obtained by \( Y^1_M \) are larger than those obtained by \( Y^1_{MR} \), which are, in turn, larger than those obtained by \( Y^1_R \). However, computing \( Y^1_M \) and \( Y^1_{MR} \) requires some a priori information about the demand distributions.

We use the step size parameter \( \alpha^k_j = (n - j + 1) \frac{200}{r_n(10 + k)} \) in Algorithms 1-3. This choice of step size parameters results in a bit more aggressive updates for the protection levels for the fare classes that arrive earlier. As evident from our backward induction in the proof of Proposition 5, the protection levels for the fare classes that arrive earlier tend to converge more slowly than those for the fare classes that arrive later. This choice of step size parameters ensures that the step size parameters that we use for the fare classes that arrive earlier do not get too small prematurely. For the method proposed by van Ryzin and McGill (2000), we use the step size parameter \( \alpha^k_j = \frac{200}{r_n(10 + k)} \), which is essentially the same step size parameter used by van Ryzin and McGill (2000) in their original paper. To be precise, van Ryzin and McGill (2000) use the step size parameter \( \alpha^k_j = \frac{200}{10 + k} \), but we scale their step size parameters and step directions respectively by \( 1/r_n \) and \( r_n \), in which case their method remains unchanged. Using a step size parameter that depends on the fare class does not affect the performance of the method proposed by van Ryzin and McGill (2000) in a systematic fashion.

We label our test problems by \( (n, \kappa) \in \{4, 8, 12\} \times \{0.95, 1.25\} \), where \( n \) is the number of fare classes and \( \kappa \) is the ratio of the total expected demand to the initial capacity. Since we test three strategies to choose the initial protection levels, this gives us 18 test cases to consider. We use the randomized version of the method proposed by van Ryzin and McGill (2000) as a benchmark strategy, since the iterates of this method converge to the optimal protection levels w.p.1 even when the demand random variables are discrete. We refer to this method as RA. We refer to Algorithms 1, 2 and 3 respectively as A1, A2 and A3. We run each method for 100 iterations on 25 sample paths and present the average results over 25 sample paths. We use common random numbers when comparing the performances of different methods.

Figure 3 compares the performances of A1 and A3 for a few test cases. The two data series in the charts plot the total expected revenues corresponding to the protection levels obtained by A1 and A3 as a function of the iteration counter. We normalize the total expected revenues by dividing by the
optimal total expected revenue, which we obtain by solving the dynamic programming formulation of
the problem. Figure 3 indicates that the performance of A3 is indistinguishable from that of A1. (In
Figure 3, the data series corresponding to A1 and A3 essentially coincide and it is difficult to see that
there are actually two data series in this figure.) Since A3 is applicable when the demand information
is censored, whereas A1 is applicable only when the relaxed view of demand censorship described in
Section 6.2 is possible, we drop A1 from further consideration. We only compare RA, A2 and A3 in the
subsequent discussion.

Figures 4, 5 and 6 respectively show the results for the test problems with 4, 8 and 12 fare classes.
In these figures, the charts on the left, in the middle and on the right respectively correspond to the
cases where the initial protection levels are $Y^1_{R}$, $Y^1_{M}$ and $Y^1_{MR}$. The top rows correspond to the cases
where $\kappa = 0.95$, whereas the bottom rows correspond to the cases where $\kappa = 1.25$. The thick, dashed
and thin data series in the charts respectively plot the total expected revenues corresponding to the
protection levels obtained by RA, A2 and A3 as a function of the iteration counter.

In the figures, the performances of A2 and A3 are very close to each other with A2 slightly lagging
from behind. The performance of A3 is almost always better than that of RA when the total expected
demand exceeds the capacity. If the total expected demand is below the capacity and the initial
protection levels are $Y^1_{M}$ or $Y^1_{MR}$, then the performance of RA is slightly better than that of A3.

We note that the protection levels $Y^1_{M}$ and $Y^1_{MR}$ are relatively good since the total expected revenues
obtained by these protection levels are about 94-98% of the optimal total expected revenues. Therefore,
the figures indicate that if the initial protection levels are good, then RA and A3 have comparable
performances. RA has a slight edge when the total expected demand is below the capacity, whereas
A3 has a slight edge when the total expected demand exceeds the capacity. On the other hand, the
protection levels $Y^1_{R}$ are not very good since the total expected revenues obtained by these protection
levels...
Figure 4: Numerical results for the test problems with 4 fare classes.

Figure 5: Numerical results for the test problems with 8 fare classes.
levels are about 71-75% of the optimal total expected revenues. In this case, the figures indicate that if the initial protection levels are not very good, then RA can lag behind A3 by a significant margin. For example, in the bottom rows of Figures 5 and 6, it appears that RA cannot get a good set of protection levels within a reasonable number of iterations when the initial protection levels are $Y_{R}^1$. To make sure that RA does not prematurely stop making progress, we run RA for 1000 iterations for these two test cases and Figure 7 plots the total expected revenues corresponding to the protection levels obtained by RA as a function of the iteration counter. RA eventually obtains good protection levels but this may take a large number of iterations.

Our numerical experiments indicate that the performances of A1, A2 and A3 are at least comparable to that of RA. There are some problem instances with tight capacities where the performance gap between the methods that we present in this paper and RA is significant. Since A1 has a convergence guarantee for the performance of the policy, it appears to be a good substitute for RA when the relaxed view of demand censorship described in Section 6.2 is possible. Despite the fact that it does not have

Figure 6: Numerical results for the test problems with 12 fare classes.

Figure 7: Performance of RA on test problems $(8, 1.25)$ and $(12, 1.25)$ when the initial protection levels are $Y_{R}^1$. 

Our numerical experiments indicate that the performances of A1, A2 and A3 are at least comparable to that of RA. There are some problem instances with tight capacities where the performance gap between the methods that we present in this paper and RA is significant. Since A1 has a convergence guarantee for the performance of the policy, it appears to be a good substitute for RA when the relaxed view of demand censorship described in Section 6.2 is possible. Despite the fact that it does not have
a convergence guarantee, A3 also seems to perform quite well and this method is applicable when the demand information is censored. Finally, we note that A1, A2 and A3 provide stochastic subgradients of the value functions with respect to the seat availability, which may be useful when making tactical decisions such as setting the capacity of the flight leg.

8 Conclusions

In this paper, we developed a stochastic approximation method to compute the optimal protection levels for the seat allocation problem under the assumption that the demand distributions are discrete. Although the problem that we consider is nonsmooth and the total expected revenue is not concave when viewed as a function of the protection levels, we were able to show that the iterates of our method converge to a set of optimal protection levels. We provided alternative versions of our method that remain applicable when the demand information is censored. Numerical experiments demonstrated that our methods are especially advantageous when the total expected demand exceeds the capacity by a significant margin and the initial protection levels are not close to the optimal protection levels.

A Appendix: Obtaining a Stochastic Subgradient of $\mathbb{E}\{v_j(\cdot, D_j)\}$

In this section, we use induction over the fare classes to show that the recursion in (10) gives a stochastic subgradient of $\mathbb{E}\{v_j(\cdot, D_j)\}$ at $x_j$. It is easy to show the result for fare class $n$. Assuming that the result holds for fare class $j + 1$ and $\dot{v}_j(x_j, d_j)$ is defined as in (10), we show that

$$v_j(\tilde{x}_j, d_j) - v_j(x_j, d_j) \leq \dot{v}_j(x_j, d_j) [\tilde{x}_j - x_j]$$

for all $x_j, \tilde{x}_j \in [0, c]$. Since the roles of $x_j$ and $\tilde{x}_j$ are interchangeable, we consider six cases.

**Case 1.** Assume that $y_j^* < x_j - d_j$ and $\tilde{x}_j - d_j \leq y_j^* \leq \tilde{x}_j$. By the induction assumption, we have

$$\mathbb{E}\{v_{j+1}(y_j^*, D_{j+1})\} - \mathbb{E}\{v_{j+1}(x_j - d_j, D_{j+1})\} \leq \mathbb{E}\{\dot{v}_{j+1}(x_j - d_j, D_{j+1})\} [y_j^* - x_j + d_j].$$

Since $y_j^*$ is a maximizer of $f_j(\cdot)$ over $[0, c]$ and $y_j^* < x_j - d_j$, we also have $\mathbb{E}\{\dot{v}_{j+1}(x_j - d_j, D_{j+1})\} \leq r_j$. Noting that $\tilde{x}_j - d_j - y_j^* \leq 0$, we obtain $r_j [\tilde{x}_j - d_j - y_j^*] \leq \mathbb{E}\{\dot{v}_{j+1}(x_j - d_j, D_{j+1})\} [\tilde{x}_j - d_j - y_j^*]$. By (9), we have

$$v_j(\tilde{x}_j, d_j) - v_j(x_j, d_j) = r_j [\tilde{x}_j - y_j^*] + \mathbb{E}\{v_{j+1}(y_j^*, D_{j+1})\} - r_j d_j - \mathbb{E}\{v_{j+1}(x_j - d_j, D_{j+1})\} \leq \mathbb{E}\{\dot{v}_{j+1}(x_j - d_j, D_{j+1})\} [\tilde{x}_j - d_j + y_j^*] + \mathbb{E}\{\dot{v}_{j+1}(x_j - d_j, D_{j+1})\} [y_j^* - x_j + d_j] = \dot{v}_j(x_j, d_j) [\tilde{x}_j - x_j],$$

where the inequality follows from the two inequalities that we derive at the beginning of this case and the last equality is by (10).

**Case 3.** Assume that $y_j^* < x_j - d_j$ and $\tilde{x}_j < y_j^*$. By the induction assumption, we have

$$\mathbb{E}\{v_{j+1}(y_j^*, D_{j+1})\} - \mathbb{E}\{v_{j+1}(x_j - d_j, D_{j+1})\} \leq \mathbb{E}\{\dot{v}_{j+1}(x_j - d_j, D_{j+1})\} [y_j^* - x_j + d_j].$$

Since $y_j^*$ is a maximizer of $f_j(\cdot)$ over $[0, c]$, we have $\mathbb{E}\{v_{j+1}(\tilde{x}_j, D_{j+1})\} - \mathbb{E}\{v_{j+1}(y_j^*, D_{j+1})\} \leq r_j [\tilde{x}_j - y_j^*]$. Adding these two inequalities, we obtain

$$\mathbb{E}\{v_{j+1}(\tilde{x}_j, D_{j+1})\} - \mathbb{E}\{v_{j+1}(x_j - d_j, D_{j+1})\} \leq \mathbb{E}\{\dot{v}_{j+1}(x_j - d_j, D_{j+1})\} [y_j^* - x_j + d_j] + r_j [\tilde{x}_j - y_j^*].$$

Similar to Case 2, since $y_j^*$ is a maximizer of $f_j(\cdot)$ over $[0, c]$ and $y_j^* < x_j - d_j,$
we also have $E\{\tilde{v}_{j+1}(x_j - d_j, D_{j+1})\} \leq r_j$. In this case, by (9), we have $v_j(\tilde{x}_j, d_j) - v_j(x_j, d_j) = E\{v_{j+1}(\tilde{x}_j, D_{j+1})\} - r_j d_j - E\{v_{j+1}(x_j - d_j, D_{j+1})\} \leq -r_j d_j + E\{\tilde{v}_{j+1}(x_j - d_j, D_{j+1})\} [y_j^* - x_j + d_j^*. + r_j [\tilde{x}_j - y_j^*] \leq E\{v_{j+1}(x_j - d_j, D_{j+1})\} [\tilde{x}_j - x_j] = \tilde{v}_j(x_j, d_j) [\tilde{x}_j - x_j]$, where the first inequality follows from the inequality that we derive at the beginning of this case, the second inequality follows from the fact that $E\{\tilde{v}_{j+1}(x_j - d_j, D_{j+1})\} \leq r_j, \tilde{x}_j < y_j^* and d_j \geq 0$, and the last equality is by (10).

**Case 4.** Assume that $x_j - d_j \leq y_j^* \leq x_j$ and $\tilde{x}_j - d_j \leq y_j^* \leq \tilde{x}_j$. By (9) and (10), we have $v_j(\tilde{x}_j, d_j) - v_j(x_j, d_j) = r_j [\tilde{x}_j - x_j] = \tilde{v}_j(x_j, d_j) [\tilde{x}_j - x_j]$.

**Case 5.** Assume that $x_j - d_j \leq y_j^* \leq x_j$ and $\tilde{x}_j < y_j^*$. Since $y_j^*$ is a maximizer of $f_j(\cdot)$ over $[0, c]$, we have $E\{v_{j+1}(\tilde{x}_j, D_{j+1})\} - E\{v_{j+1}(y_j^*, D_{j+1})\} \leq r_j [\tilde{x}_j - y_j^*]$. By (9), we have $v_j(\tilde{x}_j, d_j) - v_j(x_j, d_j) = E\{v_{j+1}(\tilde{x}_j, D_{j+1})\} - r_j [x_j - y_j^*] - E\{v_{j+1}(y_j^*, D_{j+1})\} \leq r_j [\tilde{x}_j - x_j] = \tilde{v}_j(x_j, d_j) [\tilde{x}_j - x_j]$, where the last equality is by (10).

**Case 6.** Assume that $x_j < y_j^*$ and $\tilde{x}_j < y_j^*$. We have $v_j(\tilde{x}_j, d_j) - v_j(x_j, d_j) = E\{v_{j+1}(\tilde{x}_j, D_{j+1})\} - E\{v_{j+1}(x_j, D_{j+1})\} \leq E\{v_{j+1}(x_j, D_{j+1})\} [\tilde{x}_j - x_j] = \tilde{v}_j(x_j, d_j) [\tilde{x}_j - x_j]$, where the first equality is by (9), the inequality is by the induction assumption and the second equality is by (10).

**B Appendix: Proof of Corollary 8**

We consider the seat allocation problem in (19). Using a computation similar to that in (2) and (3), we obtain

$$\tilde{v}_j(x_j, d_j) = \left\{ \max_{|x_j - d_j| \leq y_j} -r_j y_j + E\{\tilde{v}_{j+1}(y_j, D_{j+1})\} \right\} + r_j x_j. \quad (25)$$

Therefore, the optimal policy is characterized by a set of protection levels $\{\tilde{y}_j : j = 1, \ldots, n\}$, where $\tilde{y}_j^*$ can be computed as a maximizer of the function

$$\tilde{f}_j(y_j) = -r_j y_j + E\{\tilde{v}_{j+1}(y_j, D_{j+1})\} \quad (26)$$

over the interval $[0, c]$. As mentioned before, letting $\tilde{Y}_j^* = \text{argmax}_0 \leq y_j \leq c \tilde{f}_j(y_j)$, it is enough to show that $\tilde{Y}_j^* \subset [L_j^*, U_j^* + n - j]$ for all $j = 1, \ldots, n$. In Lemma 10 below, we give sufficient conditions that ensure that we have $\tilde{Y}_j^* \subset [L_j^*, U_j^* + n - j]$ for all $j = 1, \ldots, n$. After this, in Lemmas 11 and 12, we show that these sufficient conditions are indeed satisfied and this completes the proof. In Lemmas 10-12, we use $\tilde{f}_j^+(y_j)$ and $\tilde{f}_j^-(y_j)$ to respectively denote the right and left derivatives of the function $f_j(\cdot)$ at $y_j$. These derivatives exist since $f_j(\cdot)$ is concave.

**Lemma 10** Letting $f_j(\cdot)$ and $\tilde{f}_j(\cdot)$ be respectively as in (4) and (26), if we have

$$\tilde{f}_j^+(y_j) \geq f_j^+(y_j) \quad \text{for all } y_j \in [0, c) \quad (27)$$

$$\tilde{f}_j^-(y_j) \leq f_j^-(y_j - n + j) \quad \text{for all } y_j \in (n - j, c], \quad (28)$$

then we have $\tilde{Y}_j^* \subset [L_j^*, U_j^* + n - j]$ for all $j = 1, \ldots, n$. 

24
We show the result by induction over the fare classes. In particular, assuming that the result holds for \( x_j \leq y_j^* \) such that \( y_j^* \leq y_j^* \). Consequently, we determine the result holds trivially.

If, on the other hand, we have \( y_j^* > n - j \), then we obtain \( \hat{f}_j (y_j^* - n + j) \geq \hat{f}_j (y_j^*) \geq 0 \) by (28), which implies that there exists \( y_j^* \in \mathcal{Y}_j^* \) such that \( y_j^* \geq y_j^* - n + j \). Consequently, we obtain \( U_j^* \geq y_j^* - n + j \). Therefore, we have \( \hat{y}_j^* \in [L_j^*, U_j^* + n - j] \) and \( \hat{y}_j^* \in \bar{Y}_j^* \) and this implies that \( \mathcal{Y}_j^* \subset [L_j^*, U_j^* + n - j] \).

The cases where \( \hat{y}_j^* = 0 \) or \( \hat{y}_j^* = c \) can be handled in a similar manner. \( \square \)

**Lemma 11** We have \( \hat{f}_j (y_j^*) \geq \hat{f}_j (y_j^*) \) for all \( y_j \in [0, c] \), \( j = 1, \ldots, n \).

**Proof** By (4) and (9), we have

\[
v_j(x_j, d_j) = \begin{cases} f_j(x_j - d_j) + r_j x_j & \text{if } y_j^* < x_j - d_j \\ f_j(y_j^*) + r_j x_j & \text{if } x_j - d_j \leq y_j^* \leq x_j \\ f_j(x_j) + r_j x_j & \text{if } x_j < y_j^* \end{cases}
\]

which implies that

\[
\hat{v}_j(x_j, d_j) = \begin{cases} \hat{f}_j (x_j - d_j) + r_j & \text{if } y_j^* \leq x_j - d_j \\ r_j & \text{if } x_j - d_j < y_j^* \leq x_j \\ \hat{f}_j (x_j) + r_j & \text{if } x_j < y_j^* \end{cases}
\]

(29)

where \( \hat{v}_j(x_j, d_j) \) is the right derivative of the function \( v_j(\cdot, d_j) \) at \( x_j \). Similarly, by (25) and (26), we have

\[
\hat{v}_j(x_j, d_j) = \begin{cases} \hat{f}_j (x_j - d_j - 1) + r_j & \text{if } y_j^* \leq x_j - d_j - 1 \\ r_j & \text{if } x_j - d_j - 1 < y_j^* \leq x_j \\ \hat{f}_j (x_j) + r_j & \text{if } x_j < y_j^* \end{cases}
\]

(30)

On the other hand, since \( y_j^* \) and \( \hat{y}_j^* \) are respectively maximizers of \( f_j(\cdot) \) and \( \hat{f}_j(\cdot) \) over \([0, c]\), we have

\[
\hat{f}_j (x_j) \leq 0 \quad \text{for all } x_j \in [y_j^*, c) \quad (31)
\]

\[
\hat{f}_j (x_j) \geq 0 \quad \text{for all } x_j \in [0, \hat{y}_j^*). \quad (32)
\]

We show the result by induction over the fare classes. In particular, assuming that the result holds for fare class \( j \), we show that \( \hat{v}_j(x_j, d_j) \geq \hat{v}_j(x_j, d_j) \) for all \( x_j \in [0, c] \) and all \( d_j \geq 0 \). This result, together with (4) and (26), implies that \( \hat{f}_j(y_j) \geq \hat{f}_{j-1}(y_j) \) for all \( y_j \in [0, c] \) and the result follows.

Since \( \hat{f}_j(y_j) \geq \hat{f}_j (y_j) \) for all \( y_j \in [0, c] \) by the induction assumption, we can choose \( \hat{y}_j^* \) and \( y_j^* \) such that \( \hat{y}_j^* \geq y_j^* \). To show that \( \hat{v}_j(x_j, d_j) \geq \hat{v}_j(x_j, d_j) \), we consider the cases listed in Table 2.

**Case 1.** We have \( \hat{v}_j^+(x_j, d_j) = \hat{f}_j^+(x_j - d_j) + r_j \) and \( \hat{v}_j^+(x_j, d_j) = \hat{f}_j^+(x_j - d_j - 1) + r_j \) by (29) and (30). Since the right derivative of a concave function is decreasing, we have \( \hat{v}_j^+(x_j, d_j) = \hat{f}_j^+(x_j - d_j) + r_j \leq \hat{v}_j^+(x_j, d_j) \).
then we have access to hand, min

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>We have ( \dot{v}_j^+(x_j, d_j) = \dot{f}_j^-(x_j - d_j - 1) + r_j ) and ( \dot{v}_j^+(x_j, d_j) = r_j ) by (29) and (30). In this case, we have ( \dot{v}_j^+(x_j, d_j) = \dot{f}_j^+(x_j - d_j) + r_j \leq r_j = \dot{v}_j^+(x_j, d_j) ) by (31).</td>
<td></td>
</tr>
<tr>
<td>This case is the same as Case 2.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Case 5.</th>
</tr>
</thead>
<tbody>
<tr>
<td>We have ( \dot{v}_j^+(x_j, d_j) = \dot{f}_j^+(x_j - d_j - 1) + r_j ) and ( \dot{v}_j^+(x_j, d_j) = r_j ) by (29) and (30). In this case, we have ( \dot{v}_j^+(x_j, d_j) = \dot{f}_j^+(x_j - d_j) + r_j \leq r_j = \dot{v}_j^+(x_j, d_j) ) by (31).</td>
</tr>
<tr>
<td>The other five cases can be handled in the same manner.</td>
</tr>
</tbody>
</table>

Lemma 12. We have \( \tilde{f}_j(y_j) \leq \tilde{f}_j(y_j - n + j) \) for all \( y_j \in (n - j, \infty) \). |

Proof. The proof follows from an induction argument similar to the one in the proof of Lemma 11. □

### C Appendix: Proof of Corollary 9

Assume that we have access to \( \{y_j^k : j = 1, \ldots, n\}, \{x_j^k : j = 1, \ldots, n\} \) and \( \{\min\{x_j^k - \mathcal{O}(y_j^k)\}^+, D_j^k\} : j = 1, \ldots, n\}. If \( \min\{x_j^k - \mathcal{O}(y_j^k)\}^+, D_j^k\} < x_j^k - \mathcal{O}(y_j^k)^+ \), then we deduce that \( D_j^k = \min\{x_j^k - \mathcal{O}(y_j^k)\}^+, D_j^k\}. Therefore, we know the value of \( D_j^k \) and we can compute \( \min\{x_j^k - \mathcal{O}(y_j^k)\}^+, D_j^k\} = [x_j^k - \mathcal{O}(y_j^k)]^+ + 1 \leq D_j^k + 1 = \bar{d}_j^k \). If, on the other hand, \( \min\{x_j^k - \mathcal{O}(y_j^k)\}^+, D_j^k\} = [x_j^k - \mathcal{O}(y_j^k)]^+ \), then we have \( x_j^k - \mathcal{O}(y_j^k)^+ + 1 \leq D_j^k + 1 = \bar{d}_j^k \). Therefore, we deduce that \( \min\{x_j^k - \mathcal{O}(y_j^k)\}^+, D_j^k\} = [x_j^k - \mathcal{O}(y_j^k)]^+ + 1 \). This argument shows that if we have access to \( \{y_j^k : j = 1, \ldots, n\}, \{x_j^k : j = 1, \ldots, n\} \) and \( \min\{x_j^k - \mathcal{O}(y_j^k)\}^+, D_j^k\} : j = 1, \ldots, n\}, then we have access to \( \{\min\{x_j^k - \mathcal{O}(y_j^k)\}^+, D_j^k\} : j = 1, \ldots, n\}. Proposition 6 implies that if we have access to \( \{x_j^k : j = 1, \ldots, n\}, \{x_j^k : j = 1, \ldots, n\} \) and \( \min\{x_j^k - \mathcal{O}(y_j^k)\}^+, D_j^k\} : j = 1, \ldots, n\}, then we can compute \( s_j^k(y_j^k, \bar{d}_{j+1}^k, \ldots, \bar{d}_n^k) \) for all \( j = 1, \ldots, n \). □
References


Using Lagrangian Relaxation to Compute Capacity-Dependent Bid-Prices in Network Revenue Management

Huseyin Topaloglu
School of Operations Research and Industrial Engineering,
Cornell University, Ithaca, NY 14853, USA
topaloglu@orie.cornell.edu

April 1, 2006

Abstract
We propose a method to compute bid-prices in the network revenue management problem. The novel aspect of our method is that it explicitly considers the temporal dynamics of the problem and generates bid-prices that are dependent on the remaining leg capacities. Our method is based on relaxing certain constraints that link the decisions for different flight legs by associating Lagrange multipliers with them. In this case, the network revenue management problem decomposes by the flight legs and we can concentrate on one flight leg at a time. When compared with the so-called deterministic linear program, we show that our method provides a tighter upper bound on the optimal objective value of the network revenue management problem. Furthermore, computational experiments indicate that the solutions obtained by our method perform noticeably better than the ones obtained by the deterministic linear program.
The idea of bid-prices forms a powerful tool for solving network revenue management problems. This idea associates a bid-price with each flight leg, capturing the opportunity cost of a unit of capacity on the flight leg. An itinerary request is accepted only if the revenue from the itinerary request exceeds the sum of the bid-prices associated with the flight legs that are in the requested itinerary; see Williamson (1992), Talluri and van Ryzin (1998), Talluri and van Ryzin (2004). Traditionally, bid-prices are computed by solving a deterministic linear program. However, this linear program tends to be somewhat crude in the sense that it only uses the expected numbers of the future itinerary requests and completely ignores the temporal dynamics of the arrivals of the itinerary requests.

In this paper, we propose a method to compute bid-prices that explicitly considers the temporal dynamics. Our method is based on the following simple observation. The network revenue management problem is difficult because if we accept an itinerary request, then we have to consume the capacity on every flight leg that is in the requested itinerary. We relax this requirement by using Lagrangian relaxation. In particular, we allow ourselves to individually accept or reject the flight legs in a requested itinerary. When we allow such “partially-accepted” itineraries, the problem decomposes by the flight legs and we can concentrate on one flight leg at a time. This approach provides a method to compute bid-prices that explicitly considers the temporal dynamics. Furthermore, the bid-prices computed in this fashion depend on the remaining leg capacities, which is a feature lacking in the existing methods.

Our work builds on previous research. Hawkins (2003) and Adelman and Mersereau (2004) develop a Lagrangian relaxation method for what they call as loosely-coupled dynamic programs. In these dynamic programs, the evolutions of the different components of the state variable are affected by different types of decisions and these different types of decisions interact through a set of linking constraints. They propose relaxing these linking constraints by associating Lagrange multipliers with them. In this paper, we show that the network revenue management problem can be viewed as a loosely-coupled dynamic program. The Lagrangian relaxation method in Hawkins (2003) and Adelman and Mersereau (2004) runs into computational difficulties when applied to the revenue management problem. In particular, this method requires finding a good set of Lagrange multipliers by minimizing the so-called dual function and the dual function may involve thousands of dimensions. We show that it is possible to minimize the dual function efficiently by using standard subgradient optimization.

Network revenue management has been an active research area within the past decade. Talluri and van Ryzin (2004) provide a comprehensive coverage of the subject. The idea of bid-prices date back to Simpson (1989) and Williamson (1992), where the authors use the deterministic linear program mentioned above to compute bid-prices. Talluri and van Ryzin (1998) give a careful analysis of bid-prices and point out that the idea of bid-prices is equivalent to using linear value function approximations in the dynamic programming formulation of the network revenue management problem. Bertsimas and Popescu (2003) tighten the connections between bid-prices and approximate dynamic programming, and consider value function approximations that are more complicated than linear value function approximations. Adelman (2005) develops a method to compute bid-prices that explicitly considers the temporal dynamics. Although the definition of the bid-price policies given in Talluri and van Ryzin (1998) indicates that one can have bid-prices that are dependent on the remaining leg capacities, to our
knowledge, there does not exist computationally tractable methods to compute such bid-prices. Our method fills this gap.

In this paper, we make the following research contributions. 1) We develop a new method to compute bid-prices. Our method explicitly considers the temporal dynamics and generates bid-prices that depend on the remaining leg capacities. 2) Our method is based on relaxing certain constraints by associating Lagrange multipliers with them. We show that we can efficiently find a good set of Lagrange multipliers by using standard subgradient optimization. 3) We show that our method provides an upper bound on the optimal objective value of the network revenue management problem. A well-known method to obtain such an upper bound is to use the aforementioned deterministic linear program. We show that the upper bound provided by our approach is tighter than the one provided by the deterministic linear program. 4) Computational experiments indicate that our method provides noticeably better solutions than the deterministic linear program.

The rest of the paper is organized as follows. Section 1 formulates the network revenue management problem as a dynamic program. Section 2 describes the Lagrangian relaxation idea and shows how to use standard subgradient optimization to a find a good set of Lagrange multipliers. In this section, we also show that our method provides tighter upper bounds than does the deterministic linear program and the bid-prices it generates are dependent on the remaining leg capacities. Section 3 presents computational experiments.

1 Problem Formulation

We have a set of flight legs that can be used to satisfy the itinerary requests that arrive randomly over time. At each time period, an itinerary request arrives into the system and we have to decide whether to accept or reject this itinerary request. An accepted itinerary request generates a revenue and consumes the capacities on the relevant flight legs. A rejected itinerary request simply leaves the system.

The problem takes place over the finite horizon \( T = \{1, \ldots, \tau\} \) and all flight legs depart at time period \( \tau + 1 \). The set of flight legs is \( L \) and the set of itineraries is \( J \). When we accept a request for itinerary \( j \), we earn a revenue of \( f_j \) and consume \( a_{ij} \) units of capacity on flight leg \( i \). If itinerary \( j \) does not use flight leg \( i \), then we have \( a_{ij} = 0 \). The capacity on flight leg \( i \) is \( c_{i1} \). The probability that a request for itinerary \( j \) arrives into the system at time period \( t \) is \( p_{jt} \), where we assume that \( \sum_{j \in J} p_{jt} = 1 \) for all \( t \in T \). If there is a positive probability of having no itinerary requests at time period \( t \), then we can handle this by defining a dummy itinerary \( j \) with \( a_{ij} = 0 \) for all \( i \in L \) and \( f_j = 0 \). Throughout the paper, we do not differentiate between column and row vectors. We use \( |A| \) to denote the cardinality of set \( A \).

We let \( x_{it} \) be the remaining capacity on flight leg \( i \) at time period \( t \) so that \( x_t = \{x_{it} : i \in L\} \) captures the remaining leg capacities at time period \( t \). We capture the decisions at time period \( t \) by \( u_t = \{u_{jt} : j \in J\} \), where \( u_{jt} \) takes value 1 if we accept a request for itinerary \( j \) at time period \( t \), and 0 otherwise. Since our ability to accept an itinerary request is limited by the remaining leg capacities,
the set of feasible decisions at time period \( t \) can be written as

\[
U(x_t) = \{ u_t \in \{0,1\}^{\vert J \vert} : a_{ij} u_{jt} \leq x_{it} \quad \forall i \in L, \ j \in J \}.
\]

Using \( x_t \) as the state variable at time period \( t \), we can formulate the problem as a dynamic program. Letting \( C = \max\{c_{i1} : i \in L\} \) and \( \mathcal{L} = \{0,1,\ldots,C\} \), since the remaining capacity on any flight leg at any time period is less than or equal to \( C \), we use \( \mathcal{C}^{\vert L \vert} \) as the state space. In this case, the optimal policy can be found by computing the value functions through the optimality equation

\[
V_t(x_t) = \max_{u_t \in U(x_t)} \left\{ \sum_{j \in J} p_{jt} \left\{ f_j u_{jt} + V_{t+1}(x_t - u_{jt} \sum_{i \in L} a_{ij} e_i) \right\} \right\},
\]

where \( e_i \) is the \( \vert L \vert \)-dimensional unit vector with a 1 in the element corresponding to \( i \in L \). Given the state variable \( x_t \), it is easy to show that the optimal decisions at time period \( t \) are given by

\[
u^*_t(x_t) = \begin{cases} 1 & \text{if } f_j + V_{t+1}(x_t - \sum_{i \in L} a_{ij} e_i) \geq V_{t+1}(x_t) \text{ and } a_{ij} \leq x_{it} \text{ for all } i \in L \smallskip \text{and } j \in J \\ 0 & \text{otherwise}; \end{cases}
\]

see Adelman (2005). The main difficulty in solving the optimality equation in (1) arises from the fact that we have to make an “all or nothing” decision. Specifically, we either accept the itinerary request, in which case the capacities on all relevant flight legs are consumed, or we reject the itinerary request, in which case the leg capacities do not change. Our solution method is based on relaxing this requirement. That is, when an itinerary request arrives, we allow ourselves to accept or reject the individual flight legs. We make this idea precise in the following section.

2 LAGRANGIAN RELAXATION

In this section, we develop a solution method that is based on the idea of accepting or rejecting the individual flight legs.

We begin by introducing some new notation. We augment \( L \) by a fictitious flight leg \( \bar{i} \) with infinite capacity. We extend the decisions at time period \( t \) as \( y_t = \{ y_{ijt} : i \in L \cup \{\bar{i}\}, \ j \in J \} \), where \( y_{ijt} \) takes value 1 if we accept flight leg \( i \) when a request for itinerary \( j \) arrives at time period \( t \), and 0 otherwise.

In this case, it is easy to see that the optimality equation

\[
V_i(x_t) = \max_{y_t \in U(x_t)} \sum_{j \in J} p_{jt} \left\{ f_j y_{ijt} + V_{t+1}(x_t - \sum_{i \in L} y_{ijt} a_{ij} e_i) \right\}
\]

subject to

\[
a_{ij} y_{ijt} \leq x_{it} \quad \forall i \in L, \ j \in J
\]

\[
y_{ijt} - y_{ijt} = 0 \quad \forall i \in L, \ j \in J
\]

\[
y_{ijt} \in \{0,1\} \quad \forall i \in L, \ j \in J
\]

is equivalent to the optimality equation in (1). Since the capacity on the fictitious flight leg is infinite, we do not keep track of it in our state variable and the state variable in the dynamic program above is still \( x_t = \{x_{it} : i \in L\} \).

4
In the feasible solution set of problem (3)-(6), only constraints (5) link the different flight legs. This suggests associating the Lagrange multipliers \( \lambda = \{ \lambda_{ijt} : i \in \mathcal{L}, j \in \mathcal{J}, t \in \mathcal{T} \} \) with these constraints and solving the dynamic program

\[
V_t^\lambda(x_t) = \max \sum_{j \in \mathcal{J}} p_{jt} \left\{ [f_j - \sum_{i \in \mathcal{L}} \lambda_{ijt} y_{ijt} + \sum_{i \in \mathcal{L}} \lambda_{ijt} y_{ijt} + V_{t+1}^\lambda(x_{t+1} - \sum_{i \in \mathcal{L}} y_{ijt} a_{ij})] \right\}
\]

subject to (4), (6)

\[
y_{ijt} \in \{0,1\} \quad \forall j \in \mathcal{J},
\]

where we scale the Lagrange multipliers by \( \{ p_{jt} : j \in \mathcal{J}, t \in \mathcal{T} \} \) for notational clarity. If we have \( p_{jt} = 0 \) for some itinerary \( j \), then the decision variables \( \{ y_{ijt} : i \in \mathcal{L} \cup \{ i \} \} \) are inconsequential and scaling the Lagrange multipliers in this fashion does not create a complication. We use the superscript \( \lambda \) in the value functions to emphasize that the solution to the optimality equation in (7)-(9) depends on the Lagrange multipliers. We note that constraints (9) would be redundant in problem (3)-(6), but we add them to problem (7)-(9) to tighten the relaxation.

Letting \( y_{it} = \{ y_{ijt} : j \in \mathcal{J} \} \), we define the set

\[
\mathcal{Y}_{it}(x_{it}) = \{ y_{it} \in \{0,1\}^{\mathcal{J}} : a_{ij} y_{ijt} \leq x_{it} \quad \forall j \in \mathcal{J} \},
\]

in which case constraints (4) and (6) can succinctly be written as \( y_{it} \in \mathcal{Y}_{it}(x_{it}) \) for all \( i \in \mathcal{L} \). The following proposition shows that the optimality equation in (7)-(9) decomposes by the flight legs.

**Proposition 1** If \( \{ \vartheta_{i,t}^\lambda(x_{it}) : x_{it} \in \mathcal{C}, t \in \mathcal{T} \} \) is a solution to the optimality equation

\[
\vartheta_{i,t}^\lambda(x_{it}) = \max_{y_{it} \in \mathcal{Y}_{it}(x_{it})} \sum_{j \in \mathcal{J}} p_{jt} \left\{ \sum_{i \in \mathcal{L}} \lambda_{ijt} y_{ijt} + \vartheta_{i,t+1}^\lambda(x_{it} - a_{ij} y_{ijt}) \right\}
\]

for all \( i \in \mathcal{L} \), then we have

\[
V_t^\lambda(x_t) = \sum_{t' = t}^{\tau} \sum_{j \in \mathcal{J}} p_{jt'} [f_j - \sum_{i \in \mathcal{L}} \lambda_{ijt'}]^+ + \sum_{i \in \mathcal{L}} \vartheta_{i,t}^\lambda(x_{it})
\]

for all \( x_t \in \mathcal{C}^{|\mathcal{L}|}, t \in \mathcal{T} \), where we let \( [x]^+ = \max\{0,x\} \).

**Proof** We show the result by induction over the time periods. It is easy to show the result for the last time period. Assuming that the result holds for time period \( t + 1 \), problem (7)-(9) can be written as

\[
V_t^\lambda(x_t) = \max \sum_{j \in \mathcal{J}} p_{jt} \left\{ [f_j - \sum_{i \in \mathcal{L}} \lambda_{ijt} y_{ijt} + \sum_{i \in \mathcal{L}} \lambda_{ijt} y_{ijt} + \sum_{i \in \mathcal{L}} \vartheta_{i,t+1}^\lambda(x_{it} - a_{ij} y_{ijt})] \right\}
\]

subject to \( y_{it} \in \mathcal{Y}_{it}(x_{it}) \) \quad \forall i \in \mathcal{L}

\[
y_{ijt} \in \{0,1\} \quad \forall j \in \mathcal{J}.
\]
We can drop the decision variables \( \{y_{jt} : j \in J\} \) by letting \( y_{jt} = 1(f_j - \sum_{i \in L} \lambda_{ijt} \geq 0) \) for all \( j \in J \), where \( 1(\cdot) \) is the indicator function. The result follows by noting that the objective function and the feasible solution set of the problem above decomposes by the flight legs.

Therefore, we can efficiently solve the optimality equation in (7)-(9) by concentrating on one flight leg at a time. The following proposition shows that we obtain upper bounds on the value functions by solving the optimality equation in (7)-(9).

**Proposition 2** We have \( V_t(x_t) \leq V_t^\lambda(x_t) \) for all \( x_t \in \mathcal{C}[t], t \in T \).

**Proof** We show the result by induction over the time periods. It is easy to show the result for the last time period. We assume that the result holds for time period \( t + 1 \) and let \( \{\hat{y}_{jt} : i \in L \cup \{t\}, j \in J\} \) be an optimal solution to problem (3)-(6). We have

\[
V_t(x_t) = \sum_{j \in J} p_{jt} \left\{ f_j \hat{y}_{jt} + V_{t+1}(x_t - \sum_{i \in L} \hat{y}_{ijt} a_{ij} e_i) \right\}
\]

\[
= \sum_{j \in J} p_{jt} \left\{ [f_j - \sum_{i \in L} \lambda_{ijt}] \hat{y}_{jt} + \sum_{i \in L} \lambda_{ijt} \hat{y}_{ijt} + V_{t+1}(x_t - \sum_{i \in L} \hat{y}_{ijt} a_{ij} e_i) \right\}
\]

\[
\leq \sum_{j \in J} p_{jt} \left\{ [f_j - \sum_{i \in L} \lambda_{ijt}] \hat{y}_{jt} + \sum_{i \in L} \lambda_{ijt} \hat{y}_{ijt} + V_{t+1}^\lambda(x_t - \sum_{i \in L} \hat{y}_{ijt} a_{ij} e_i) \right\} \leq V_t^\lambda(x_t),
\]

where the second equality follows from (5), the first inequality follows from the induction hypothesis, and the second inequality follows from the fact that \( \hat{y}_{ijt} \in \{0,1\} \) for all \( j \in J \) and \( \hat{y}_{jt} = \{\hat{y}_{jt} : j \in J\} \in Y_{it}(x_{it}) \) for all \( i \in L \).

Letting \( c_1 = \{c_{i1} : i \in L\} \), Proposition 2 implies that \( V_1(c_1) \leq V_1^\lambda(c_1) \). Noting that \( V_1(c_1) \) is the maximum total expected revenue obtained over the time periods \( \{1, \ldots, \tau\} \), we can obtain a tight bound on this quantity by solving

\[
\min_\lambda \left\{ V_1^\lambda(c_1) \right\} = \min_\lambda \left\{ \sum_{t \in T} \sum_{j \in J} p_{jt} [f_j - \sum_{i \in L} \lambda_{ijt}]^+ + \sum_{i \in L} \theta^\lambda_{i1}(c_{i1}) \right\}, \tag{11}
\]

where the equality follows from Proposition 1. The objective function of problem (11) is called the dual function. In the following section, we show that problem (11) can be solved efficiently.

### 2.1 Minimizing the Dual Function

In this section, we show that \( \theta^\lambda_{i1}(c_{i1}) \) is a convex function of \( \lambda \) and its subgradients can be computed by solving the optimality equation in (10). This allows us to solve problem (11) by using standard subgradient optimization.

We begin by introducing some new notation. We let \( \{y_{ijt}^\lambda(x_{it}) : j \in J\} \) be an optimal solution to problem (10), where the superscript \( \lambda \) and the argument \( x_{it} \) indicate that the optimal solution depends on the Lagrange multipliers and the remaining leg capacity. In this case, (10) can be written as

\[
\theta^\lambda_{it}(x_{it}) = \sum_{j \in J} p_{jt} \left\{ \lambda_{ijt} y_{ijt}^\lambda(x_{it}) + \sum_{x_{i,t+1} \in \mathcal{C}} 1(x_{i,t+1} = x_{it} - a_{ij} y_{ijt}^\lambda(x_{it})) \vartheta_{i,t+1}(x_{i,t+1}) \right\}, \tag{12}
\]
To write the expression above in matrix notation, we let $Y_{it}^\lambda$ be the $|C| \times |J|$-dimensional matrix whose $(x_{it}, j)$-th element is $p_{jt} y_{ijt}(x_{it})$ and $Q_{it}^\lambda$ be the $|C| \times |C|$-dimensional matrix whose $(x_{it}, x_{i,t+1})$-th element is $\sum_{j \in J} p_{jt} 1(x_{i,t+1} = x_{it} - a_{ij} y_{ijt}(x_{it}))$. Letting $\vartheta_{it}^\lambda$ be the vector $\{\vartheta_{it}^\lambda(x_{it}) : x_{it} \in C\}$ and $\lambda_{it}$ be the vector $\{\lambda_{ijt} : j \in J\}$, (12) can be written in matrix notation as

$$\vartheta_{it}^\lambda = Y_{it}^\lambda \lambda_{it} + Q_{it}^\lambda \vartheta_{i,t+1}^\lambda.$$ 

We are now ready to show that $\vartheta_{i1}^\lambda(c_{i1})$ is a convex function of $\lambda$.

**Proposition 3** For any two sets of Lagrange multipliers $\lambda$ and $\lambda^0$, we have

$$\vartheta_{it}^{\lambda^0} \geq \vartheta_{it}^{\lambda} + Y_{it}^\lambda [\lambda_{it}^0 - \lambda_{it}] + Q_{it}^\lambda Y_{i,t+1}^\lambda [\lambda_{i,t+1}^0 - \lambda_{i,t+1}] + \ldots + Q_{it}^\lambda Q_{i,t+1}^\lambda \ldots Q_{i,\tau-1}^\lambda Y_{i\tau}^\lambda [\lambda_{i\tau}^0 - \lambda_{i\tau}]$$

for all $i \in \mathcal{L}$, $t \in \mathcal{T}$.

**Proof** We show the result by induction over the time periods. It is easy to show the result for the last time period. We assume that the result holds for time period $t + 1$. Since $\{y_{ijt}(x_{it}) : j \in J\}$ is an optimal solution to problem (10), we have

$$\vartheta_{it}^\lambda(x_{it}) = \sum_{j \in J} p_{jt} \left\{ \lambda_{ijt} y_{ijt}(x_{it}) + \sum_{x_{i,t+1} \in C} 1(x_{i,t+1} = x_{it} - a_{ij} y_{ijt}(x_{it})) \vartheta_{i,t+1}^\lambda(x_{i,t+1}) \right\}$$

$$\vartheta_{it}^{\lambda^0}(x_{it}) \geq \sum_{j \in J} p_{jt} \left\{ \lambda_{ijt}^0 y_{ijt}(x_{it}) + \sum_{x_{i,t+1} \in C} 1(x_{i,t+1} = x_{it} - a_{ij} y_{ijt}(x_{it})) \vartheta_{i,t+1}^{\lambda^0}(x_{i,t+1}) \right\}.$$ 

Subtracting the first expression from the second one, the resulting expression can be written in matrix notation as

$$\vartheta_{it}^{\lambda^0} - \vartheta_{it}^\lambda \geq Y_{it}^\lambda [\lambda_{it}^0 - \lambda_{it}] + Q_{it}^\lambda [\vartheta_{i,t+1}^{\lambda^0} - \vartheta_{i,t+1}^\lambda].$$

(13) The result follows by using the induction assumption that

$$\vartheta_{i,t+1}^{\lambda^0} \geq \vartheta_{i,t+1}^\lambda + Y_{i,t+1}^\lambda [\lambda_{i,t+1}^0 - \lambda_{i,t+1}] + Q_{i,t+1}^\lambda Y_{i,t+2}^\lambda [\lambda_{i,t+2}^0 - \lambda_{i,t+2}]$$

$$+ \ldots + Q_{i,t+1}^\lambda Q_{i,t+2}^\lambda \ldots Q_{i,\tau-1}^\lambda Y_{i\tau}^\lambda [\lambda_{i\tau}^0 - \lambda_{i\tau}]$$

in (13) and noting that the matrix $Q_{it}^\lambda$ has positive elements. \hfill $\Box$

Letting $\Pi_{it}^\lambda = Q_{i1}^\lambda Q_{i2}^\lambda \ldots Q_{i,t-1}^\lambda Y_{i\tau}^\lambda$ with $\Pi_{i1}^\lambda = Y_{i1}^\lambda$, we have

$$\vartheta_{i1}^{\lambda^0} \geq \vartheta_{i1}^\lambda + \Pi_{i1}^\lambda [\lambda_{i1}^0 - \lambda_{i1}] + \Pi_{i2}^\lambda [\lambda_{i2}^0 - \lambda_{i2}] + \ldots + \Pi_{i\tau}^\lambda [\lambda_{i\tau}^0 - \lambda_{i\tau}]$$

by Proposition 3. Letting $\alpha_{i1}$ be the $|C|$-dimensional unit vector with a 1 in the $c_{i1}$-th element, we obtain

$$\vartheta_{i1}^{\lambda^0}(c_{i1}) = \alpha_{i1} \vartheta_{i1}^{\lambda^0} \geq \alpha_{i1} \vartheta_{i1}^\lambda + \sum_{t \in T} \alpha_{i1} \Pi_{it}^\lambda [\lambda_{it}^0 - \lambda_{it}] = \vartheta_{i1}^\lambda(c_{i1}) + \sum_{t \in T} \alpha_{i1} \Pi_{it}^\lambda [\lambda_{it}^0 - \lambda_{it}].$$

Therefore, $\vartheta_{i1}^\lambda(c_{i1})$ has a subgradient and Theorem 3.2.6 in Bazaraa, Sherali and Shetty (1993) implies that $\vartheta_{i1}^\lambda(c_{i1})$ is a convex function of $\lambda$. The dual function, being a sum of convex functions of $\lambda$, is also a convex function of $\lambda$ and we can use standard subgradient optimization to solve problem (11).
2.2 Relationship with the Deterministic Linear Program

An alternative solution method for the problem described in Section 1 is to solve a deterministic linear program. Letting $w_j$ be the number of requests for itinerary $j$ that we plan to accept, this linear program has the form

$$\text{max} \quad \sum_{j \in J} f_j w_j$$

subject to

$$\sum_{j \in J} a_{ij} w_j \leq c_i \quad \forall i \in L$$

$$w_j \leq \sum_{t \in T} p_{jt} \quad \forall j \in J$$

$$w_j \geq 0 \quad \forall j \in J. \quad (17)$$

Constraints (15) ensure that the numbers of itinerary requests that we plan to accept do not violate the leg capacities. Constraints (16) ensure that we do not plan to accept more itinerary requests than the expected numbers of itinerary requests.

There are two main uses of problem (14)-(17). First, this problem can be used to decide whether we should accept or reject an itinerary request. In particular, letting $\{\mu^*_i : i \in L\}$ be the optimal values of the dual variables associated with constraints (15), the idea is to use $\mu^*_i$ as an estimate of the marginal value of the capacity on flight leg $i$. In the network revenue management vocabulary, this estimate of the marginal value is called the bid-price for the flight leg. If the revenue from an itinerary request exceeds the sum of the bid-prices of the flight legs that are in the requested itinerary, then we accept the itinerary request. Specifically, if we have

$$f_j \geq \sum_{i \in L} a_{ij} \mu^*_i, \quad (18)$$

then we accept a request for itinerary $j$. Letting $\tilde{V}_t(x_t) = \sum_{i \in L} \mu^*_i x_{it}$ for all $i \in L$, $t \in T$ and noting that $\tilde{V}_{t+1}(x_t) - \tilde{V}_{t+1}(x_t - \sum_{i \in L} a_{ij} x_{it}) = \sum_{i \in L} a_{ij} \mu^*_i$, it is easy to see that (18) is equivalent to approximating the value functions $\{V_t(x_t) : x_t \in \mathcal{C}^{|L|}, t \in T\}$ in (2) by $\{\tilde{V}_t(x_t) : x_t \in \mathcal{C}^{|L|}, t \in T\}$. This approach is simple to implement and the computational experiments in Williamson (1992) indicate that it provides good solutions.

Second, it is possible to show that the optimal objective value of problem (14)-(17) provides an upper bound on the maximum total expected revenue obtained over the time periods $\{1, \ldots, \tau\}$; see Bertsimas and Popescu (2003). That is, letting $\zeta^*$ be the optimal objective value of problem (14)-(17), we have $V_1(c_1) \leq \zeta^*$. This information can be useful when assessing the optimality gap of a suboptimal policy such as the one in (18).

In the remainder of this section, we show that we have

$$V_1(c_1) \leq \min_{\lambda} \left\{ V_1^\lambda(c_1) \right\} \leq \zeta^*. \quad (19)$$

Therefore, we can obtain a tighter upper bound on $V_1(c_1)$ by solving problem (11). Since the first inequality above follows from Proposition 2, we concentrate only on the second inequality.
Using the decision variables \( \{ z_{ij} : i \in L \cup \{ i \}, j \in J \} \), we write problem (14)-(17) as

\[
\begin{align*}
\text{max} & \quad \sum_{j \in J} f_j z_{ij} \quad (20) \\
\text{subject to} & \quad \sum_{j \in J} a_{ij} z_{ij} \leq c_{i1} \quad \forall i \in L \quad (21) \\
& \quad z_{ij} \leq \sum_{t \in T} p_{jt} \quad \forall j \in J \quad (22) \\
& \quad z_{ij} - z_{ij} = 0 \quad \forall i \in L, j \in J \quad (23) \\
& \quad z_{ij}, z_{ij} \geq 0 \quad \forall i \in L, j \in J. \quad (24)
\end{align*}
\]

By duality theory, there exist Lagrange multipliers \( \{ \beta^o_{ij} : i \in L, j \in J \} \) such that the problem

\[
\begin{align*}
\text{max} & \quad \sum_{j \in J} \left[ f_j - \sum_{i \in L} \beta^o_{ij} \right] z_{ij} + \sum_{i \in L} \sum_{j \in J} \beta^o_{ij} z_{ij} \\
\text{subject to} & \quad (21), (22), (24)
\end{align*}
\]

has the same optimal objective value as problem (20)-(24). Noting the upper and lower bounds on the decision variables \( \{ z_{ij} : j \in J \} \), the problem above becomes

\[
\begin{align*}
\text{max} & \quad \sum_{t \in T} \sum_{j \in J} p_{jt} \left[ f_j - \sum_{i \in L} \beta^o_{ij} \right] z_{ij} + \sum_{i \in L} \sum_{j \in J} \beta^o_{ij} z_{ij} \\
\text{subject to} & \quad (21), (22), (24)
\end{align*}
\]

Letting \( \{ \mu_i : i \in L \} \) be the dual variables associated with constraints (26), the dual of problem (25)-(27) can be written as

\[
\begin{align*}
\text{min} & \quad \sum_{t \in T} \sum_{j \in J} p_{jt} \left[ f_j - \sum_{i \in L} \beta^o_{ij} \right] + \sum_{i \in L} c_{i1} \mu_i \\
\text{subject to} & \quad (21), (22), (24)
\end{align*}
\]

Therefore, problem (28)-(30) has the same optimal objective value as problem (14)-(17). We are now ready to show that (19) holds.

**Proposition 4** We have \( \min_\lambda \left\{ V_1^\lambda(c_1) \right\} \leq \zeta^* \).

**Proof** We let \( \{ \mu^*_i : i \in L \} \) be an optimal solution to problem (28)-(30) and define the Lagrange multipliers \( \lambda^o = \{ \lambda^o_{ijt} : i \in L, j \in J, t \in T \} \) as \( \lambda^o_{ijt} = \beta^o_{ij} \) for all \( i \in L, j \in J, t \in T \). We begin by using induction over the time periods to show that \( \vartheta^\lambda_{ijt}(x_{it}) \leq \mu^*_i x_{it} \) for all \( x_{it} \in C, i \in L, t \in T \). For the last time period, we have

\[
\vartheta^\lambda_{ijt}(x_{it}) = \max_{y_{it} \in Y_{it}(x_{it})} \left\{ \sum_{j \in J} p_{jt} \beta^o_{ij} y_{ijt} \right\} \leq \max_{y_{it} \in Y_{it}(x_{it})} \left\{ \sum_{j \in J} p_{jt} \mu^*_i a_{ijy_{ijt}} \right\} \leq \sum_{j \in J} p_{jt} \mu^*_i x_{it} = \mu^*_i x_{it}.
\]
where the first inequality follows from (29) and the second inequality follows from the fact that \( y_{it} \in Y_{it}(x_{it}) \). Therefore, the result holds for the last time period. Assuming that the result holds for time period \( t+1 \), we have

\[
\vartheta^\lambda(x_{it}) = \max_{y_{it} \in Y_{it}(x_{it})} \left\{ \sum_{j \in J} p_{jt} \left\{ \beta_{ij} y_{ijt} + \vartheta^\lambda_{i,t+1}(x_{it} - a_{ij} y_{ijt}) \right\} \right\}
\]

\[
\leq \max_{y_{it} \in Y_{it}(x_{it})} \left\{ \sum_{j \in J} p_{jt} \left\{ \beta_{ij} y_{ijt} + \mu^*_{i} [x_{it} - a_{ij} y_{ijt}] \right\} \right\}
\]

\[
= \max_{y_{it} \in Y_{it}(x_{it})} \left\{ \sum_{j \in J} p_{jt} \left[ \beta_{ij} - \mu^*_i a_{ij} \right] y_{ijt} \right\} + \mu^*_i x_{it} \leq \mu^*_i x_{it},
\]

where the first inequality follows from the induction hypothesis and the second inequality follows from (29). This establishes that \( \vartheta^\lambda(x_{it}) \leq \mu^*_i x_{it} \) for all \( x_{it} \in \mathcal{C}, \ i \in \mathcal{L}, \ t \in \mathcal{T} \). In particular, we have \( \vartheta^\lambda(c_{i1}) \leq \mu^*_i c_{i1} \) for all \( i \in \mathcal{L} \), which implies that

\[
\min_{\lambda} \left\{ V_1^\lambda(c_1) \right\} \leq V_1^{\lambda^*}(c_1) = \sum_{t \in T} \sum_{j \in J} p_{jt} \left[ f_j - \sum_{i \in \mathcal{L}} \beta_{ij}^0 \right]^+ + \sum_{i \in \mathcal{L}} \vartheta_{i1}^{\lambda^*}(c_{i1}) \leq \sum_{t \in T} \sum_{j \in J} p_{jt} \left[ f_j - \sum_{i \in \mathcal{L}} \beta_{ij}^0 \right]^+ + \sum_{i \in \mathcal{L}} \mu^*_i c_{i1} = \zeta^*,
\]

where the first equality follows from Proposition 1 and the second equality follows by noting the objective function of problem (28)-(30).

### 2.3 Bid-Price Structure of the Greedy Policy

Letting \( \lambda^* \) be an optimal solution to problem (11), our solution method approximates the value functions \( \{V_t(x_t) : x_t \in \mathcal{C}^{[t]}, \ t \in \mathcal{T}\} \) in (2) by \( \{V_t^{\lambda^*}(x_t) : x_t \in \mathcal{C}^{[t]}, \ t \in \mathcal{T}\} \). Specifically, given the state variable \( x_t \), if we have

\[
f_j + V_{t+1}^{\lambda^*}(x_t) - \sum_{i \in \mathcal{L}} a_{ij} e_i \geq V_{t+1}^{\lambda^*}(x_t),
\]

then we accept a request for itinerary \( j \) at time period \( t \).

It is easy to see that this idea leads to bid-prices similar to those in (18). Using Proposition 1, (31) can be written as

\[
f_j \geq \sum_{i \in \mathcal{L}} \sum_{r=1}^{a_{ij}} \left\{ \vartheta_{i,t+1}^{\lambda^*}(x_{it} + 1 - r) - \vartheta_{i,t+1}^{\lambda^*}(x_{it} - r) \right\}.
\]

In this case, we can view \( \vartheta_{i,t+1}^{\lambda^*}(x_{it}) - \vartheta_{i,t+1}^{\lambda^*}(x_{it} - 1) \) as the bid-price for the \( x_{it} \)-th unit of capacity on flight leg \( i \). Similar to (18), if the revenue from an itinerary request exceeds the sum of the bid-prices of the flight legs that are in the requested itinerary, then we accept the itinerary request. However, the bid-price of a flight leg is constant in (18), whereas the bid-price of a flight leg depends on the remaining leg capacity in (32).
3 Computational Experiments

This section compares the performance of our solution method with the performance of a standard benchmark strategy.

3.1 Benchmark Strategy and Experimental Setup

We use the solution method described in Section 2.2 as a benchmark strategy. In particular, to make the decisions at time period $t$, we replace the right side of constraints (16) by \( \{ \sum_{t'=t}^{\tau} p_{j'} : j \in J \} \) and solve problem (14)-(17). Letting \( \{ \mu_i^* : i \in L \} \) be the optimal values of the dual variables associated with constraints (15), if we have $f_j \geq \sum_{i \in L} a_{ij} \mu_i^*$, then we accept a request for itinerary $j$ at time period $t$. We refer to this solution method as LP.

We use two variants of our solution method. In the first variant, we solve problem (11) to obtain an optimal solution $\lambda^*$. Given the state variable $x_t$, if we have $f_j + V_{t+1}^\lambda(x_t - \sum_{i \in L} a_{ij} e_i) \geq V_{t+1}^\lambda(x_t)$, then we accept a request for itinerary $j$ at time period $t$. In the second variant, given the state variable $x_t$, we solve the problem $\min_{\lambda} \{ V_1^\lambda(c_1) \}$ to obtain an optimal solution $\lambda^*_t$ and if we have $f_j + V_{t+1}^\lambda(x_t - \sum_{i \in L} a_{ij} e_i) \geq V_{t+1}^\lambda(x_t)$, then we accept a request for itinerary $j$ at time period $t$. Therefore, the second variant refines the value function approximations at each time period. We refer to the first and second variants of our solution method respectively as SLG and DLG. Since DLG refines the value function approximations at each time period, we expect it to perform better than SLG, but this is not guaranteed.

We use standard subgradient optimization to solve the problems $\min_{\lambda} \{ V_1^\lambda(c_1) \}$ and $\min_{\lambda} \{ V_t^\lambda(x_t) \}$; see Wolsey (1998). We initialize the step size parameter to $\sum_{j \in J} f_j / |J|$, and double the step size parameter after each iteration that results in a decrease in the objective function value and halve the step size parameter after each iteration that results in an increase in the objective function value. Although it does not guarantee convergence to an optimal solution, adjusting the step size parameter in this fashion provides good solutions and stable performance.

Our test problems consider an airline network that serves $N$ locations out of a single hub; this is a key network structure that frequently arises in practice. There are two flight legs associated with each location, one to the hub and one from the hub. There is a high-fare and a low-fare itinerary that connects each origin-destination pair. Consequently, we have $2N$ flight legs and $2N(N+1)$ itineraries, $4N$ of which involve one flight leg and $2N(N-1)$ of which involve two flight legs. The revenues associated with the high-fare itineraries are four times larger than the revenues associated with the low-fare itineraries. The probability of having a request for a high-fare itinerary increases over time, whereas the probability of having a request for a low-fare itinerary decreases over time. Since $\sum_{t \in T} \sum_{j \in J} p_{jt} a_{ij}$ is the total expected demand for the capacity on flight leg $i$, we measure the tightness of the leg capacities by

$$\alpha = \frac{\sum_{t \in T} \sum_{j \in J} p_{jt} a_{ij}}{\sum_{i \in L} c_i}.$$  \hspace{1cm} (33)

In all of our test problems, we have $\tau = 200$. We label our test problems by the pair $(N, \alpha) \in \{2, 4, 8\} \times \{0.9, 1.0, 1.2, 1.6\}$, where $N$ is the number of locations and $\alpha$ is as in (33).
Problem \((N, \alpha)\) | \(\zeta^*\) | \(V_1^+(c_1)\) | \(V_1^+(c_1)/\zeta^*\) | CPU | LP | LGS | LGD | LGD/LP | % LGD\geq LP
---|---|---|---|---|---|---|---|---|---|
(2,0.9) | 11,123 | 10,991 | 0.99 | 26 | 10,937 | 10,968 | 10,972 | 1.00 | 69
(4,0.9) | 21,562 | 21,135 | 0.98 | 256 | 20,802 | 20,760 | 20,891 | 1.00 | 55
(8,0.9) | 20,107 | 19,527 | 0.97 | 46 | 18,630 | 18,517 | 18,990 | 1.02 | 86
(2,1.0) | 11,114 | 10,655 | 0.96 | 38 | 10,497 | 10,603 | 10,615 | 1.01 | 75
(4,1.0) | 21,531 | 20,439 | 0.95 | 376 | 19,780 | 19,805 | 20,054 | 1.01 | 77
(8,1.0) | 20,052 | 18,975 | 0.95 | 190 | 17,694 | 17,415 | 18,257 | 1.03 | 96
(2,1.2) | 10,163 | 9,775 | 0.96 | 19 | 9,475 | 9,733 | 9,754 | 1.03 | 88
(4,1.2) | 19,882 | 18,938 | 0.95 | 364 | 17,873 | 17,962 | 18,433 | 1.03 | 94
(8,1.2) | 18,952 | 17,472 | 0.92 | 1,002 | 16,264 | 16,711 | 16,996 | 1.04 | 95
(2,1.6) | 8,827 | 8,440 | 0.96 | 19 | 8,033 | 8,410 | 8,425 | 1.05 | 96
(4,1.6) | 17,530 | 16,600 | 0.95 | 312 | 15,345 | 15,536 | 16,019 | 1.04 | 93
(8,1.6) | 16,833 | 15,295 | 0.91 | 830 | 13,974 | 14,594 | 14,753 | 1.06 | 94

Table 1: Computational results.

### 3.2 Computational Results

We summarize our computational results in Table 1. In this table, the second and third columns respectively give the optimal objective values of problems (14)-(17) and (11). The fifth column gives the CPU seconds required to solve problem (11) on a Pentium IV Desktop PC with 2.4 GHz CPU and 1 GB RAM running Windows XP. The sixth, seventh and eighth columns respectively give the total expected revenues obtained by LP, LGS and LGD. We estimate these total expected revenues by simulating 100 trajectories. The tenth column gives in what percent of the 100 simulated trajectories the revenues obtained by LGD are equal to or better than those obtained by LP.

The second and third columns indicate that problem (11) provides up to 9% tighter upper bounds on the maximum total expected revenue than does problem (14)-(17). The sixth, seventh and eighth columns indicate that LGD performs better than LP and LGS. The performance gap between LGD and LP can be up to 6% when the leg capacities are tight. (We use common random numbers for simulating the trajectories for different solution methods. The performance gap between LGD and LP is statistically significant at 1% level for all test problems; see Law and Kelton (2000).) The tenth column indicates that the revenues obtained by LGD are at least as good as those obtained by LP for a majority of the simulated trajectories. This is especially the case for the test problems with large numbers of locations and tight leg capacities. Finally, it is interesting to note that the performance of LGS is comparable to that of LP.

### References


A Stochastic Approximation Method to Compute Bid Prices in Network Revenue Management Problems

Huseyin Topaloglu
School of Operations Research and Industrial Engineering,
Cornell University, Ithaca, New York 14853, USA
topaloglu@orie.cornell.edu

January 10, 2007

Abstract

We present a stochastic approximation method to compute bid prices in network revenue management problems. The key idea is to visualize the total expected revenue as a function of the bid prices and to use sample path-based derivatives to search for a good set of bid prices. We deal with the discrete nature of the network revenue management setting by formulating a smoothed version of the problem, which assumes that it is possible to accept a fraction of an itinerary request. We show that the iterates of our method converge to a stationary point of the total expected revenue function of the smoothed version. Computational experiments demonstrate that the bid prices obtained by our method outperform the ones obtained by standard benchmark methods and our method is especially advantageous when the bid prices are not recomputed frequently.
The notion of bid prices forms a powerful tool to construct good policies for network revenue management problems. The idea is to associate a bid price with each flight leg that captures the opportunity cost of a unit of capacity. An itinerary request is accepted if and only if there is enough capacity and the revenue from the itinerary request exceeds the sum of the bid prices associated with the flight legs in the requested itinerary; see Williamson (1992) and Talluri and van Ryzin (2004).

Bid prices are traditionally computed by solving a deterministic linear program. However, this linear program only uses the expected numbers of the itinerary requests that are to arrive until the departure time and does not attempt to capture the probability distributions or temporal dynamics of the arrivals of the itinerary requests. In this paper, we propose a new stochastic approximation method to compute bid prices. The key idea is to visualize the total expected revenue as a function of the bid prices and to search for a good set of bid prices by using sample path-based derivatives. Since the sample path-based derivatives of the total revenue function depend on what itinerary requests arrive in what order, our stochastic approximation method explicitly captures the temporal dynamics of the arrivals of the itinerary requests.

Focusing on a class of policies characterized by a number of parameters and utilizing stochastic approximation methods to search for a good set of values for the parameters is a common approach in stochastic optimization. Nevertheless, this approach, combined with the discrete nature of the decisions in the network revenue management setting, brings unique challenges. First, if we perturb the bid price of a flight leg by an infinitesimal amount, then the number of itinerary requests that we accept can only change by an integer amount, which implies that the sample path-based derivative of the total revenue function either does not exist or is equal to zero. We deal with this difficulty by formulating a smoothed version of the problem. The smoothed version assumes that the leg capacities are continuous and we can accept a fraction of an itinerary request depending on how much the revenue from the itinerary request exceeds the sum of the bid prices associated with the flight legs in the requested itinerary. It is, of course, not realistic to assume that we can accept a fraction of an itinerary request and we drop this assumption when implementing the policy in practice. Second, even in the smoothed version of the problem, the sample path-based derivative of the total revenue function does not exist when there are multiple flight legs whose capacities are simultaneously binding. We deal with this difficulty by perturbing the leg capacities by small random amounts. These two modifications ensure that the sample path-based derivative of the total revenue function exists with probability one (w.p.1) and we can develop a convergent stochastic approximation method to search for a good set of bid prices.

Our work draws on two papers in particular. Mahajan and van Ryzin (2001) propose a stochastic approximation method to choose stocking levels for substitutable products. Since the total profit is not a concave function of the stocking levels, they formulate a smoothed version of their problem by assuming that the stocking levels can take fractional values and this allows them to develop a convergent stochastic approximation method. Our smoothed version is inspired by their work. On the other hand, van Ryzin and Vulcano (2006) propose a stochastic approximation method to choose protection levels in network revenue management problems. They show that the sample path-based derivative of the total revenue function does not exist when the protection level of a virtual class coincides with the capacity.
on the flight leg. They resolve this difficulty by perturbing the leg capacities by small random amounts whenever they make a decision. Our perturbations of the leg capacities are inspired by their work.


The literature on computing bid prices in network revenue management problems is also related to our work. Simpson (1989) and Williamson (1992) were the first to use the aforementioned deterministic linear program to compute bid prices. Talluri and van Ryzin (1998) give a careful study of the policies characterized by bid prices and show that these policies are asymptotically optimal as the leg capacities and the expected numbers of the itinerary requests increase linearly with the same rate. Talluri and van Ryzin (1999) propose a randomized version of the deterministic linear program that uses actual samples of the numbers of the itinerary requests as opposed to expected values. Bertsimas and Popescu (2003) use a method that captures the total opportunity cost of the leg capacities consumed by an itinerary request more accurately. Adelman (2006) works with the linear programming representation of the dynamic programming formulation of the network revenue management problem to compute bid prices. Topaloglu (2006) uses a relaxation strategy to decompose the network revenue management problem by the flight legs and computes bid prices by concentrating on one flight leg at a time.

We make the following research contributions in this paper. 1) We propose a new method to compute bid prices that uses sample path-based derivatives of the total revenue function. Our method does not require computing expectations explicitly and allows having arbitrary nonstationarities and correlations in the arrivals of the itinerary requests. 2) We show that the iterates of our method converge to a stationary point of the total expected revenue function. 3) Computational experiments demonstrate that the bid prices obtained by our method may outperform the ones obtained by standard benchmark methods. The performance gap becomes particularly noticeable when there are multiple fare classes with large differences in the fares. Furthermore, our method is especially advantageous when the bid prices are not recomputed frequently.

The paper is organized as follows. Section 1 formulates a basic optimization problem that maximizes the total expected revenue by choosing the bid prices. Section 2 describes a smoothed version of this problem that can be solved by using sample path-based derivatives. Section 3 gives an algorithm to solve the smoothed version and shows that the iterates of this algorithm converge to a stationary point of the total expected revenue function. Section 4 presents computational experiments.

1 Problem Formulation

We have a set of flight legs that can be used to satisfy the itinerary requests that arrive randomly over time. Whenever an itinerary request arrives, we have to decide whether to accept or reject it. An
accepted itinerary request generates a revenue and consumes the capacities on the relevant flight legs. A rejected itinerary request simply leaves the system.

We use $\mathcal{L}$ to denote the set of flight legs in the airline network and $\mathcal{J}$ to denote the set of possible itineraries. If we accept a request for itinerary $j$, then we generate a revenue of $\bar{r}_j$ and consume $\bar{a}_{ij}$ units of capacity on flight leg $i$. If flight leg $i$ is not in itinerary $j$, then we naturally have $\bar{a}_{ij} = 0$. The itinerary requests arrive sequentially and we index them by $t \in \{1, 2, \ldots\}$. We use $x_{it}$ to denote the remaining capacity on flight leg $i$ just before making the decision for itinerary request $t$. Therefore, the initial capacity on flight leg $i$ is $x_{i1}$ and $x_1 = \{x_{i1} : i \in \mathcal{L}\}$ is a part of the problem data.

Assuming that the total number of itinerary requests is bounded by a finite integer $\tau$, we characterize the arrivals of the itinerary requests by the stochastic process $\tilde{\omega} = \{J_t : t = 1, \ldots, \tau\}$. Itinerary request $t$ is for itinerary $J_t$ and the value of the random variable $J_t$ becomes known just before making the decision for itinerary request $t$. If we let $r_t = \tilde{r}_{J_t}$ and $a_{it} = \tilde{a}_{iJ_t}$ for notational brevity, then the random variables $r_t$ and $a_t = \{a_{it} : i \in \mathcal{L}\}$ capture all of the information related to itinerary request $t$. Therefore, we can alternatively characterize the arrivals of the itinerary requests by the stochastic process $\omega = \{(r_t, a_t) : t = 1, \ldots, \tau\}$. Throughout the paper, we work with the stochastic process $\omega$ rather than $\tilde{\omega}$, although these two stochastic processes are equivalent. As far as $\omega$ is concerned, we only assume that $|r_t| \leq B_r$ and $a_{it} \in \{0, \ldots, B_a\}$ w.p.1 for a finite scalar $B_r$ and a finite integer $B_a$. Other than these assumptions, $\omega$ can be a general stochastic process involving arbitrary nonstationarities and correlations among the itinerary requests. We also note that since we do not necessarily have exactly $\tau$ itinerary requests in all sample paths of $\omega$, we allow having $(r_t,a_t) = (0,0)$ for all $t = \tau^0 + 1, \ldots, \tau$ for some random variable $\tau^0$ taking values in $\{1, \ldots, \tau\}$. In this case, accepting the last $\tau - \tau^0$ itinerary requests would neither generate revenue nor consume the leg capacities.

The policy characterized by bid prices $\lambda = \{\lambda_i : i \in \mathcal{L}\}$ accepts an itinerary request if and only if there is enough capacity and the revenue from the itinerary request exceeds the sum of the bid prices associated with the flight legs in the requested itinerary. Therefore, as a function of the remaining leg capacities and itinerary requests, the decision function of this policy can be written as

$$\tilde{u}_t(x_t, \omega, \lambda) = 1(x_t \geq a_t, r_t \geq \sum_{i \in \mathcal{L}} a_{it} \lambda_i),$$

(1)

where $1(\cdot)$ is the indicator function and $x_t = \{x_{it} : i \in \mathcal{L}\}$ are the remaining leg capacities just before making the decision for itinerary request $t$. If the policy accepts itinerary request $t$, then we have $\tilde{u}_t(x_t, \omega, \lambda) = 1$. Otherwise, we have $\tilde{u}_t(x_t, \omega, \lambda) = 0$.

As a function of the remaining leg capacities and itinerary requests, the cumulative revenue function of the policy characterized by bid prices $\lambda$ can be written recursively as

$$\tilde{R}_t(x_t, \omega, \lambda) = r_t \tilde{u}_t(x_t, \omega, \lambda) + \tilde{R}_{t+1}(x_t - \tilde{u}_t(x_t, \omega, \lambda) a_t, \omega, \lambda),$$

(2)

with $\tilde{R}_{\tau+1}(\cdot, \cdot, \cdot, \lambda) = 0$. If we use the policy characterized by bid prices $\lambda$, then the total revenue that we generate from all itinerary requests is given by the random variable $\tilde{R}_1(x_1, \omega, \lambda)$. Therefore, we can find a good set of bid prices by solving the problem

$$\max_{\lambda} \mathbb{E}\{\tilde{R}_1(x_1, \omega, \lambda)\}.$$
One approach to solve this problem is to use the sample path-based derivatives of $\tilde{R}_1(x_1, \omega, \lambda)$. However, the difficulty with using the sample path-based derivatives is that if $\lambda_i$ is perturbed by an infinitesimal amount, then the result of the decision function in (1) either does not change or changes by one. This implies that the derivative of $\tilde{R}_1(x_1, \omega, \lambda)$ with respect to $\lambda_i$ is either equal to zero or does not exist, in which case it is impossible to obtain useful sample path-based derivatives. In the next section, we resolve this difficulty by formulating a smoothed version of problem (3).

2 Smoothing the Decision and Revenue Functions

In this section, we formulate a smoothed version of problem (3) that can be solved by using sample path-based derivatives. The fundamental idea behind the smoothed version is to assume that the leg capacities are continuous and we can accept a fraction of an itinerary request.

We consider a policy that accepts a fraction of an itinerary request depending on how much the revenue from the itinerary request exceeds the sum of the bid prices associated with the flight legs in the requested itinerary. For this purpose, we let $\theta(\cdot)$ be an increasing and differentiable function that satisfies $\lim_{p \to -\infty} \theta(p) = 1$ and $\lim_{p \to -\infty} \theta(p) = 0$. The policy characterized by bid prices $\lambda$ accepts $\theta(r_t - \sum_{i \in L} a_{it} \lambda_i)$ units of itinerary request $t$ as long as there is enough capacity. Therefore, the decision function of this policy can be written as

$$\hat{u}_t(x_t, \omega, \lambda) = \min \left\{ \min_{i \in \mathcal{L}_t^+} \left\{ (x_t/a_{it}) \right\}, \theta(r_t - \sum_{i \in \mathcal{L}} a_{it} \lambda_i) \right\},$$

where we let $\mathcal{L}_t^+ = \{i \in \mathcal{L} : a_{it} > 0\}$. It is easy to see that we have $\hat{u}_t(x_t, \omega, \lambda) a_{it} \leq x_{it}$ for all $i \in \mathcal{L}$ and the decision function above does not violate the leg capacities. We assume that $\theta(\cdot)$ and its derivative $\dot{\theta}(\cdot)$ are Lipschitz. That is, there exist finite scalars $L_\theta$ and $\hat{L}_\theta$ such that we have $|\theta(p) - \theta(q)| \leq L_\theta |p - q|$ and $|\dot{\theta}(p) - \dot{\theta}(q)| \leq \hat{L}_\theta |p - q|$ for all $p, q \in \mathbb{R}$. We discuss possible choices for $\theta(\cdot)$ in Section 4.

The decision function in (4) is still not differentiable with respect to $\lambda$ when $\min_{i \in \mathcal{L}_t^+} \{x_{it}/a_{it}\} = \theta(r_t - \sum_{i \in \mathcal{L}} a_{it} \lambda_i)$. To overcome this difficulty, we use an approach proposed by van Ryzin and Vulcano (2006). We let $\alpha = \{\alpha_{it} : i \in \mathcal{L}, t = 1, \ldots, \tau\}$ be uniformly distributed random variables over a small interval $[0, \epsilon]$ and perturb the leg capacities by $\alpha_t = \{\alpha_{it} : i \in \mathcal{L}\}$ just before making the decision for itinerary request $t$. Therefore, we use the decision function

$$u_t(x_t, \omega, \alpha, \lambda) = \min \left\{ \min_{i \in \mathcal{L}_t^+} \left\{ (x_{it} + \alpha_{it})/a_{it} \right\}, \theta(r_t - \sum_{i \in \mathcal{L}} a_{it} \lambda_i) \right\}.$$

Assuming that $\alpha$ is independent of $\omega$ and $\{\alpha_{it} : i \in \mathcal{L}, t = 1, \ldots, \tau\}$ are independent of each other, the event $\min_{i \in \mathcal{L}_t^+} \{x_{it} + \alpha_{it}/a_{it}\} = \theta(r_t - \sum_{i \in \mathcal{L}} a_{it} \lambda_i)$ occurs with probability zero and the decision function above is differentiable with respect to $\lambda$ w.p.1.

Similar to (2), the cumulative revenue function of the policy characterized by bid prices $\lambda$ can be written recursively as

$$R_t(x_t, \omega, \alpha, \lambda) = r_t u_t(x_t, \omega, \alpha, \lambda) + R_{t+1} (x_t + \alpha_t - u_t(x_t, \omega, \alpha, \lambda) a_t, \omega, \alpha, \lambda),$$

(6)
with $R_{t+1}(\cdot, \cdot, \cdot, \lambda) = 0$. In this case, the smoothed version of the problem that we want to solve is

$$\max_{\lambda} \mathbb{E}\{ R_1(x_1, \omega, \alpha, \lambda) \}. \quad (7)$$

Using the fact that the decision function in (5) is differentiable with respect to $\lambda$ w.p.1, one can check by backward induction on (6) that $R_1(x_1, \omega, \alpha, \lambda)$ is differentiable with respect to $\lambda$ w.p.1 and it may be possible to solve problem (7) by using the sample path-based derivatives of $R_1(x_1, \omega, \alpha, \lambda)$.

Finally, we use (5) to obtain

$$\partial^A_i R_t(x_t, \omega, \alpha, \lambda) = \frac{\partial R_t(z_t, \omega, \alpha, \gamma)}{\partial\gamma_i}\bigg|_{(z_t, \gamma) = (x_t, \lambda)}$$

and

$$\partial^X_i R_t(x_t, \omega, \alpha, \lambda) = \frac{\partial R_t(z_t, \omega, \alpha, \gamma)}{\partial z_{it}}\bigg|_{(z_t, \gamma) = (x_t, \lambda)}.$$

We also use $\partial^A_i u_t(x_t, \omega, \alpha, \lambda)$ and $\partial^X_i u_t(x_t, \omega, \alpha, \lambda)$ with similar interpretations. Differentiating (6) with respect to the bid price of flight leg $i$ and using the chain rule, we have

$$\partial^A_i R_t(x_t, \omega, \alpha, \lambda) = r_t \partial^A_i u_t(x_t, \omega, \alpha, \lambda) + \partial^A_i R_{t+1}(x_t + \alpha_t - u_t(x_t, \omega, \alpha, \lambda) a_t, \omega, \alpha, \lambda)$$

$$- \sum_{j \in \mathcal{L}} a_{jt} \partial^X_i u_t(x_t, \omega, \alpha, \lambda) \partial^X_j R_{t+1}(x_t + \alpha_t - u_t(x_t, \omega, \alpha, \lambda) a_t, \omega, \alpha, \lambda). \quad (8)$$

To compute the terms on the right side above, we use (5) to obtain

$$\partial^A_i u_t(x_t, \omega, \alpha, \lambda) = \begin{cases} -a_{it} \theta(r_t - \sum_{j \in \mathcal{L}} a_{jt} \lambda_j) & \text{if } \theta(r_t - \sum_{j \in \mathcal{L}} a_{jt} \lambda_j) \leq \min_{j \in \mathcal{L}^+} \{[x_{jt} + \alpha_{jt}]/a_{jt}\} \\ 0 & \text{otherwise.} \end{cases} \quad (9)$$

On the other hand, differentiating (6) with respect to the remaining capacity on flight leg $i$, we have

$$\partial^X_i R_t(x_t, \omega, \alpha, \lambda) = r_t \partial^X_i u_t(x_t, \omega, \alpha, \lambda)$$

$$+ \sum_{j \in \mathcal{L}} \left[ 1(j = i) - a_{jt} \partial^X_i u_t(x_t, \omega, \alpha, \lambda) \right] \partial^X_j R_{t+1}(x_t + \alpha_t - u_t(x_t, \omega, \alpha, \lambda) a_t, \omega, \alpha, \lambda). \quad (10)$$

Finally, we use (5) to obtain

$$\partial^X_i u_t(x_t, \omega, \alpha, \lambda) = \begin{cases} 1/a_{it} & \text{if } i = \text{argmin}_{j \in \mathcal{L}^+} \{[x_{jt} + \alpha_{jt}]/a_{jt}\} \text{ and } [x_{it} + \alpha_{it}]/a_{it} \leq \theta(r_t - \sum_{j \in \mathcal{L}} a_{jt} \lambda_j) \\ 0 & \text{otherwise.} \end{cases} \quad (11)$$

We can compute $\partial^A_1 R_1(x_1, \omega, \alpha, \lambda)$ simply by simulating the decisions of the policy characterized by bid prices $\lambda$ under itinerary requests $\omega$ and perturbation random variables $\alpha$. In this case, the leg capacities just before making the decision for itinerary request $t + 1$ are given recursively by

$$X_{t+1}(x_1, \omega, \alpha, \lambda) = X_t(x_1, \omega, \alpha, \lambda) + a_t - u_t(X_t(x_1, \omega, \alpha, \lambda), \omega, \alpha, \lambda) a_t,$$
with \( X_1(x_1, \omega, \alpha, \lambda) = x_1 \). Using (9) and (11), we can compute \( \partial_i^A u_t(X_t(x_1, \omega, \alpha, \lambda), \omega, \alpha, \lambda) \) and \( \partial_i^X u_t(X_t(x_1, \omega, \alpha, \lambda), \omega, \alpha, \lambda) \) for all \( i \in \mathcal{L}, t = 1, \ldots, \tau \). We can compute \( \partial_i^X R_t(X_t(x_1, \omega, \alpha, \lambda), \omega, \alpha, \lambda) \) for all \( i \in \mathcal{L}, t = 1, \ldots, \tau \) by using (10) and moving backwards through the itinerary requests. Finally, we can compute \( \partial_i^A R_t(X_t(x_1, \omega, \alpha, \lambda), \omega, \alpha, \lambda) \) for all \( i \in \mathcal{L}, t = 1, \ldots, \tau \) by using (8) and moving backwards through the itinerary requests. In the next section, we describe an algorithm that uses the sample path-based derivatives given by (8)-(11) to search for a stationary point of the objective function of problem (7).

3 Solution Algorithm and Convergence Analysis

We propose the following algorithm to solve problem (7).

**Algorithm 1**

**Step 1.** Initialize the bid prices \( \lambda^1 = \{\lambda^1_i : i \in \mathcal{L}\} \) arbitrarily and initialize the iteration counter by letting \( k = 1 \).

**Step 2.** Letting \( \omega^k \) be the itinerary requests and \( \alpha^k \) be the perturbation random variables at iteration \( k \), compute \( \partial_i^A R_1(x_1, \omega^k, \alpha^k, \lambda^k) \) for all \( i \in \mathcal{L} \) by using (8)-(11).

**Step 3.** Letting \( \sigma^k \) be a step size parameter, compute the bid prices \( \lambda^{k+1} = \{\lambda^{k+1}_i : i \in \mathcal{L}\} \) at the next iteration as \( \lambda^{k+1}_i = \lambda^k_i + \sigma^k \partial_i^A R_1(x_1, \omega^k, \alpha^k, \lambda^k) \) for all \( i \in \mathcal{L} \).

**Step 4.** Increase \( k \) by 1 and go to Step 2.

We let \( \mathcal{F}^k \) be the filtration generated by the random variables \( \{\lambda^1, \omega^1, \ldots, \omega^{k-1}, \alpha^1, \ldots, \alpha^{k-1}\} \) in this algorithm and assume that the joint distribution of \( (\omega^k, \alpha^k) \) conditional on \( \mathcal{F}^k \) is the same as the joint distribution of \( (\omega, \alpha) \). We have the next convergence result for Algorithm 1.

**Proposition 1** Assume that the sequence of step size parameters \( \{\sigma^k\}_k \) are \( \mathcal{F}^k \)-measurable and satisfy \( \sigma^k \geq 0 \) for all \( k = 1, 2, \ldots, \sum_{k=1}^{\infty} \alpha^k = \infty \) and \( \sum_{k=1}^{\infty} ||\sigma^k||^2 < \infty \) w.p.1. If the sequence of bid prices \( \{\lambda^k\}_k \) are generated by Algorithm 1, then we have \( \lim_{k \to \infty} \mathbb{E}\{\partial_i^A R_1(x_1, \omega, \alpha, \lambda^k)\} = 0 \) w.p.1 for all \( i \in \mathcal{L} \) and every limit point of the sequence of bid prices \( \{\lambda^k\}_k \) is a stationary point of the objective function of problem (7) w.p.1.

**Proof** Propositions 2, 3 and 6 below show that the following statements hold for all \( \lambda, \gamma \in \mathbb{R}^{||\mathcal{L}||_I}, i \in \mathcal{L} \).

(A.1) Using \( \partial_i^A \mathbb{E}\{R_1(x_1, \omega, \alpha, \lambda)\} \) to denote the derivative of \( \mathbb{E}\{R_1(x_1, \omega, \alpha, \cdot)\} \) with respect to the bid price of flight leg \( i \) evaluated at bid prices \( \lambda \), we have \( \mathbb{E}\{\partial_i^A R_1(x_1, \omega, \alpha, \lambda)\} = \partial_i^A \mathbb{E}\{R_1(x_1, \omega, \alpha, \lambda)\} \).

(A.2) There exists a finite scalar \( B^A_R \) such that we have \( ||\partial_i^A R_1(x_1, \omega, \alpha, \lambda)|| \leq B^A_R \) w.p.1.

(A.3) Using \( || \cdot || \) to denote the Euclidean norm, there exists a finite scalar \( L^R_R \) such that we have \( \mathbb{E}\{||\partial_i^A R_1(x_1, \omega, \alpha, \lambda) - \partial_i^A R_1(x_1, \omega, \alpha, \gamma)||\} \leq L^R_R ||\lambda - \gamma|| \).

In this case, the result follows from Proposition 4.1 in Bertsekas and Tsitsiklis (1996), which we briefly state in the appendix for completeness. In particular, since the joint distribution of \( (\omega^k, \alpha^k) \) conditional on \( \mathcal{F}^k \) is the same as the joint distribution of \( (\omega, \alpha) \), (A.1), (A.2) and (A.3) respectively show that (B.1), (B.2) and (B.3) in Proposition 4.1 in Bertsekas and Tsitsiklis (1996) are satisfied. \( \square \)
(A.1) implies that the expected value of the step direction that we use in Step 3 of Algorithm 1 is an ascent direction of the objective function of problem (7). (A.2) implies that the norm of the step direction is uniformly bounded. (A.3) implies that the expected value of the step direction is Lipschitz when viewed as a function of the bid prices. If we can show that the cumulative revenue function in (6) satisfies (A.1)-(A.3), then the iterates of Algorithm 1 converge to a stationary point of the objective function of problem (7). The next proposition shows that (A.1) holds.

**Proposition 2** We have $\mathbb{E}\{\partial^A_i R_1(x_1, \omega, \alpha, \lambda)\} = \partial^A_i \mathbb{E}\{R_1(x_1, \omega, \alpha, \lambda)\}$ for all $\lambda \in \mathbb{R}^{|\mathcal{L}|}$, $i \in \mathcal{L}$.

**Proof** Since we have $|\min\{p_1, p_2\} - \min\{q_1, q_2\}|^2 \leq |p_1 - q_1|^2 + |p_2 - q_2|^2$, $\min\{\cdot, \cdot\} : \mathbb{R}^2 \to \mathbb{R}$ is Lipschitz. Since $\theta(\cdot)$ is also Lipschitz and the composition of Lipschitz functions is Lipschitz, the decision function in (5) is Lipschitz when viewed as a function of the bid prices and leg capacities. Moving backwards through the itinerary requests and using the fact that the composition of Lipschitz functions is Lipschitz, one can show that the cumulative revenue function in (6) is Lipschitz when viewed as a function of the bid prices and leg capacities. Therefore, $R_1(x_1, \omega, \alpha, \cdot)$ is Lipschitz. By the discussion in Section 2, the derivative of $R_1(x_1, \omega, \alpha, \cdot)$ with respect to the bid price of flight leg $i$ evaluated at bid prices $\lambda$ exists w.p.1. In this case, since the cumulative revenue function is also bounded by $\tau B_r$ w.p.1, the result follows from Lemma 6.3.1 in Glasserman (1994), which we briefly state in the appendix for completeness. \hfill $\square$

The next proposition focuses on (A.2). We emphasize that the scalars mentioned in Propositions 3 and 6 and Lemmas 4 and 5 below are simple constants that do not depend on the leg capacities and bid prices.

**Proposition 3** We have $|\partial^A_i R_1(x_1, \omega, \alpha, \lambda)| \leq B^A_R$ w.p.1 for a finite scalar $B^A_R$.

**Proof** All statements in the proof are in w.p.1 sense. We first show that

$$|\partial^X_i R_t(x_t, \omega, \alpha, \lambda)| \leq B_r + B_r (1 + B_a) |\mathcal{L}| + \ldots + B_r (1 + B_a)^{\tau-t} |\mathcal{L}|^{\tau-t}$$

for all $x_t \in \mathbb{R}^{|\mathcal{L}|}_+, i \in \mathcal{L}, t = 1, \ldots, \tau$. Since $a_{it} \geq 1$ for all $i \in \mathcal{L}^+_t$, we have $|\partial^X_i u_t(x_t, \omega, \alpha, \lambda)| \leq 1$ for all $i \in \mathcal{L}$ by (11) and (10) implies that

$$|\partial^X_i R_t(x_t, \omega, \alpha, \lambda)| \leq B_r + \sum_{j \in \mathcal{L}} (1 + B_a) |\partial^X_j R_{t+1}(x_t + \alpha_t - u_t(x_t, \omega, \alpha, \lambda) a_t, \omega, \alpha, \lambda)|$$

for all $x_t \in \mathbb{R}^{|\mathcal{L}|}_+, i \in \mathcal{L}$. Using the inequality above and moving backwards through the itinerary requests, it is easy to show that (12) holds. Therefore, if we let $B^X_R = B_r + B_r (1 + B_a) |\mathcal{L}| + \ldots + B_r (1 + B_a)^{\tau-1} |\mathcal{L}|^{\tau-1}$, then we have $|\partial^X_i R_t(x_t, \omega, \alpha, \lambda)| \leq B^X_R$ for all $x_t \in \mathbb{R}^{|\mathcal{L}|}_+, i \in \mathcal{L}, t = 1, \ldots, \tau$.

Since $L_\theta$ is the Lipschitz modulus of $\theta(\cdot)$, we have $|\theta(p)| \leq L_\theta$ for all $p \in \mathbb{R}$ and (9) implies that $|\partial^A_i u_t(x_t, \omega, \alpha, \lambda)| \leq B_a L_\theta$ for all $x_t \in \mathbb{R}^{|\mathcal{L}|}_+, i \in \mathcal{L}, t = 1, \ldots, \tau$. Since $|\partial^X_i R_t(x_t, \omega, \alpha, \lambda)| \leq B^X_R$, (8) implies that $|\partial^A_i R_t(x_t, \omega, \alpha, \lambda)| \leq B_r B_a L_\theta + |\partial^A_i R_{t+1}(x_t + \alpha_t - u_t(x_t, \omega, \alpha, \lambda) a_t, \omega, \alpha, \lambda)| + B^2_a B^X_R |\mathcal{L}| L_\theta$ for all $x_t \in \mathbb{R}^{|\mathcal{L}|}_+, i \in \mathcal{L}$. Using this inequality and moving backwards through the itinerary requests, it is easy to show that $|\partial^A_i R_1(x_1, \omega, \alpha, \lambda)| \leq \tau [B_r B_a L_\theta + B^2_a B^X_R |\mathcal{L}| L_\theta]$ and the result follows. \hfill $\square$
We introduce some new notation to show that (A.3) holds. We let $x_t^\lambda$ be the leg capacities just before making the decision for itinerary request $t$ when we use the policy characterized by bid prices $\lambda$. That is, the random variables $\{x_t^\lambda : t = 1, \ldots, \tau\}$ are given recursively by

$$x_{t+1}^\lambda = x_t^\lambda + \alpha_t - u_t(x_t^\lambda, \omega, \alpha, \lambda) \alpha_t,$$

with $x_1^\lambda = x_1$. The next two lemmas are preliminary results that are useful to show that (A.3) holds. Lemma 4 shows that the expected value of the derivative of the decision function with respect to the bid price of flight leg $i$ is Lipschitz when viewed as a function of the bid prices.

**Lemma 4** We have $\mathbb{E}\left\{ |\partial^A u(t(x_t^\lambda, \omega, \alpha, \lambda) - \partial^A u_t(x_t^\lambda, \omega, \alpha, \gamma)| \right\} \leq L_u^A \|\lambda - \gamma\|$ for a finite scalar $L_u^A$.

**Proof** We consider four cases. First, we assume that $\theta(\sum_{j \in L} a_j, \lambda_j) \leq \min_{j \in \Lambda^L} \{x_j^\lambda + \alpha_jt/a_{jt}\}$ and $\theta(\sum_{j \in L} a_j, \lambda_j) \leq \min_{j \in \Lambda^L} \{x_j^\lambda + \alpha_jt/a_{jt}\}$. By (9), we have $|\partial^A u_t(x_t^\lambda, \omega, \alpha, \lambda) - \partial^A u(x_t^\lambda, \omega, \alpha, \gamma)| = a_{it} |\theta(\sum_{j \in L} a_j, \lambda_j) - \theta(\sum_{j \in L} a_j, \lambda_j)\| \leq B_\theta |\sum_{j \in L} a_j \lambda_j| \leq B_\theta |\lambda - \gamma|$. Second, we assume that $\theta(\sum_{j \in L} a_j, \lambda_j) > \min_{j \in \Lambda^L} \{x_j^\lambda + \alpha_jt/a_{jt}\}$. We have $|\partial^A u_t(x_t^\lambda, \omega, \alpha, \lambda) - \partial^A u(x_t^\lambda, \omega, \alpha, \gamma)| = a_{it} |\theta(\sum_{j \in L} a_j, \lambda_j) - \theta(\sum_{j \in L} a_j, \lambda_j)| \leq B_\theta |\lambda - \gamma|$. Third, we assume that $\theta(\sum_{j \in L} a_j, \lambda_j) < \min_{j \in \Lambda^L} \{x_j^\lambda + \alpha_jt/a_{jt}\}$. Therefore, we have a bound on the probability of the second case by noting that

$$\|x_t^\lambda - x_t^\gamma\| \leq L_X \|\lambda - \gamma\| \text{ w.p.1 for a finite scalar } L_X.$$

We now obtain a bound on the probability of the second case. Lemma 10 in the appendix shows that $\|x_t^\lambda - x_t^\gamma\| \leq L_X \|\lambda - \gamma\|$ w.p.1 for a finite scalar $L_X$. Therefore, we have

$$\left| a_{it} \theta(\sum_{j \in L} a_j, \lambda_j) - x_t^\lambda \right| \leq B_\theta \left( \theta(\sum_{j \in L} a_j, \lambda_j) - \theta(\sum_{j \in L} a_j, \lambda_j) \right) + \|x_t^\lambda - x_t^\gamma\| \leq B_\theta^2 |L_x| \|\lambda - \gamma\| + L_X \|\lambda - \gamma\| \quad (14)$$

w.p.1. On the other hand, for two sets of random variables $\{P_i : i \in A\}$ and $\{Q_i : i \in A\}$, we have $\mathbb{P}\{p \leq \min_{j \in A} \{P_j\}, q > \min_{j \in A} \{Q_j\}\} \leq \sum_{i \in A} \mathbb{P}\{p \leq \min_{j \in A} \{P_j\}, q > Q_i\} \leq \sum_{i \in A} \mathbb{P}\{p \leq P_i, q > Q_i\}$. Therefore, we obtain a bound on the probability of the second case by noting that

$$\mathbb{P}\{\theta(\sum_{j \in L} a_j, \lambda_j) \leq \min_{j \in \Lambda^L} \{x_j^\lambda + \alpha_jt/a_{jt}\}, \theta(\sum_{j \in L} a_j, \lambda_j) > \min_{j \in \Lambda^L} \{x_j^\lambda + \alpha_jt/a_{jt}\}\}$$

$$\leq \sum_{i \in L} \mathbb{P}\{\theta(\sum_{j \in L} a_j, \lambda_j) \leq x_t^\lambda + \alpha_{it}, \theta(\sum_{j \in L} a_j, \lambda_j) > x_t^\lambda + \alpha_{it}\}$$

$$= \sum_{i \in L} \mathbb{P}\{\alpha_{it} \theta(\sum_{j \in L} a_j, \lambda_j) - x_t^\lambda \leq \alpha_{it} < x_t^\lambda \theta(\sum_{j \in L} a_j, \lambda_j) - x_t^\lambda\}$$

$$\leq \left[ B_\theta^2 L_x^2 |L_x| + |L| L_X \right] \|\lambda - \gamma\|/\epsilon,$$

where the second inequality follows from (14), and the fact that $\alpha_{it}$ is uniformly distributed over the interval $[0, \epsilon]$ and it is independent of $x_t^\lambda$ and $a_{it}$. The same bound applies to the probability of the third
case. Combining the four cases at the beginning of the proof and using the trivial bound of one for the probability of the first case, we obtain
\[
\mathbb{E}\{\|\partial^i_t u_t(x^i, \omega, \alpha, \lambda) - \partial^i_t u_t(x^i, \omega, \alpha, \gamma)\|\} \leq B_a^2 |\mathcal{L}| L_\theta \|\lambda - \gamma\| + 2 [B_a^2 |\mathcal{L}|^2 L_\theta + |\mathcal{L}| L_X] \|\lambda - \gamma\| B_a L_\theta / \epsilon
\]
and the result follows. \(\square\)

Lemma 5 shows that the expected value of the derivative of the cumulative revenue function with respect to the remaining capacity on flight leg \(i\) is Lipschitz when viewed as a function of the bid prices. Its proof is similar to that of Proposition 6 and is deferred to the appendix.

**Lemma 5** We have \(\mathbb{E}\{\|\partial^X_t R_t(x^i, \omega, \alpha, \lambda) - \partial^X_t R_t(x^i, \omega, \alpha, \gamma)\|\} \leq L^X_R \|\lambda - \gamma\|\) for a finite scalar \(L^X_R\).

Finally, the next proposition shows that \((A.3)\) holds.

**Proposition 6** We have \(\mathbb{E}\{\|\partial^A_t R_t(x^i, \omega, \alpha, \lambda) - \partial^A_t R_t(x^i, \omega, \alpha, \gamma)\|\} \leq L^A_R \|\lambda - \gamma\|\) for a finite scalar \(L^A_R\).

**Proof** All statements in the proof are in w.p.1 sense. Using \((13)\), we can write \((8)\) as
\[
\partial^A_t R_t(x^i, \omega, \alpha, \lambda) = r_t \partial^A_t u_t(x^i, \omega, \alpha, \lambda) + \partial^A_t R_{t+1}(x^i_t, \omega, \alpha, \lambda)
\]
\[
- \sum_{j \in \mathcal{L}} a_{ij} \partial^A_t u_t(x^i, \omega, \alpha, \lambda) \partial^X_t R_{t+1}(x^i_t, \omega, \alpha, \lambda).
\]

On the other hand, using the fact that \(|p_1 q_1 - p_2 q_2| \leq |p_1| |q_1 - q_2| + |p_1 - p_2| |q_2|\), we have
\[
|\partial^A_t u_t(x^i, \omega, \alpha, \lambda)| \partial^X_t R_{t+1}(x^i_t, \omega, \alpha, \lambda) - \partial^A_t u_t(x^i, \omega, \alpha, \gamma)| \partial^X_t R_{t+1}(x^i_t, \omega, \alpha, \gamma)|
\]
\[
\leq |\partial^A_t u_t(x^i, \omega, \alpha, \lambda)| |\partial^X_t R_{t+1}(x^i_t, \omega, \alpha, \lambda) - \partial^X_t R_{t+1}(x^i_t, \omega, \alpha, \gamma)|
\]
\[
+ |\partial^A_t u_t(x^i, \omega, \alpha, \lambda) - \partial^A_t u_t(x^i, \omega, \alpha, \gamma)| |\partial^X_t R_{t+1}(x^i_t, \omega, \alpha, \gamma)|.\]

Using the fact that \(\|\partial^A_t u_t(x^i, \omega, \alpha, \lambda)\| \leq B_a L_\theta\) and noting \(B^X_R\) in the proof of Proposition 3, the inequality above implies that
\[
|\partial^A_t u_t(x^i, \omega, \alpha, \lambda)| \partial^X_t R_{t+1}(x^i_t, \omega, \alpha, \lambda) - \partial^A_t u_t(x^i, \omega, \alpha, \gamma)| \partial^X_t R_{t+1}(x^i_t, \omega, \alpha, \gamma)|
\]
\[
\leq B_a L_\theta |\partial^X_t R_{t+1}(x^i_t, \omega, \alpha, \lambda) - \partial^X_t R_{t+1}(x^i_t, \omega, \alpha, \gamma)|
\]
\[
+ B^X_R |\partial^A_t u_t(x^i, \omega, \alpha, \lambda) - \partial^A_t u_t(x^i, \omega, \alpha, \gamma)|. \quad (16)
\]

Therefore, by \((15)\) and \((16)\), we obtain
\[
\mathbb{E}\{\|\partial^A_t R_t(x^i, \omega, \alpha, \lambda) - \partial^A_t R_t(x^i, \omega, \alpha, \gamma)\|\}
\]
\[
\leq B_r \mathbb{E}\{\|\partial^A_t u_t(x^i, \omega, \alpha, \lambda) - \partial^A_t u_t(x^i, \omega, \alpha, \gamma)\|\}
\]
\[
+ \mathbb{E}\{\|\partial^A_t R_{t+1}(x^i, \omega, \alpha, \lambda) - \partial^A_t R_{t+1}(x^i, \omega, \alpha, \gamma)\|\}
\]
\[
+ \sum_{j \in \mathcal{L}} B_a^2 L_\theta \mathbb{E}\{\|\partial^X_t R_{t+1}(x^i_t, \omega, \alpha, \lambda) - \partial^X_t R_{t+1}(x^i_t, \omega, \alpha, \gamma)\|\}
\]
\[
+ \sum_{j \in \mathcal{L}} B_a B^X_R \mathbb{E}\{\|\partial^A_t u_t(x^i, \omega, \alpha, \lambda) - \partial^A_t u_t(x^i, \omega, \alpha, \gamma)\|\},
\]
in which case Lemmas 4 and 5 imply that
\[
\mathbb{E}\left\{ \left| \partial_t^\Lambda R_t(x_t^\lambda, \omega, \alpha, \lambda) - \partial_t^\Lambda R_t(x_t^\gamma, \omega, \alpha, \gamma) \right| \right\} \\
\leq B_r L_\Lambda \|\lambda - \gamma\| + \mathbb{E}\left\{ \left| \partial_t^\Lambda R_{t+1}(x_{t+1}^\lambda, \omega, \alpha, \lambda) - \partial_t^\Lambda R_{t+1}(x_{t+1}^\gamma, \omega, \alpha, \gamma) \right| \right\} \\
+ B_2 |\mathcal{L}| L_\theta L_\Lambda^X \|\lambda - \gamma\| + B_a B_R^X |\mathcal{L}| L_\Lambda^X \|\lambda - \gamma\|
\]

Using the inequality above and moving backwards through the itinerary requests, it is easy to show that
\[
\mathbb{E}\left\{ \left| \partial_t^\Lambda R_t(x_t^\lambda, \omega, \alpha, \lambda) - \partial_t^\Lambda R_t(x_t^\gamma, \omega, \alpha, \gamma) \right| \right\} \leq (\tau - t + 1) \left[ (B_r L_\Lambda + B_2^2 |\mathcal{L}| L_\theta L_\Lambda^X + B_a B_R^X |\mathcal{L}| L_\Lambda^X) \|\lambda - \gamma\| \right]
\]
and the result follows by letting \( L_R^\Lambda = \tau \left[ B_r L_\Lambda + B_2^2 |\mathcal{L}| L_\theta L_\Lambda^X + B_a B_R^X |\mathcal{L}| L_\Lambda^X \right] \|\lambda - \gamma\| \).  

4 Computational Experiments

In this section, we compare the performances of the bid prices obtained by solving problem (7) with the performances of the bid prices obtained by using other benchmark strategies.

4.1 Experimental Setup

In our test problems, we consider an airline network that serves \( N \) spokes out of a single hub. This is an important network structure that frequently arises in practice. Associated with each spoke, there are two flight legs, one of which is from the hub and the other one is to the hub. There is a high-fare and a low-fare itinerary that connects each origin-destination pair. Therefore, we have \( 2N \) flight legs and \( 2N(N + 1) \) itineraries, \( 4N \) of which include one flight leg and \( 2N(N - 1) \) of which include two flight legs.

The revenues associated with the high-fare itineraries are \( \rho \) times larger than the revenues associated with the low-fare itineraries. The probability of having a request for a high-fare itinerary increases over time, whereas the probability of having a request for a low-fare itinerary decreases over time. Figure 1 shows the structure of the airline network for the case where \( N = 6 \). Since the total expected demand for the capacity on flight leg \( i \) is \( \sum_{t=1}^\tau \mathbb{E}\{a_{it}\} \), we measure the tightness of the leg capacities by

\[
\kappa = \frac{\sum_{i \in \mathcal{L}} \sum_{t=1}^\tau \mathbb{E}\{a_{it}\}}{\sum_{i \in \mathcal{L}} x_{i1}}.
\]

We vary \( N, \kappa \) and \( \rho \) in our test problems and label them by \((N, \kappa, \rho) \in \{6, 12\} \times \{1.0, 1.2, 1.6\} \times \{2, 4, 8\} \). In our test problems, the initial capacities on the flight legs range over the interval \([25, 50]\), the total expected numbers of itinerary requests range over the interval \([250, 500]\) and the fares range over the interval \([50, 750]\).

4.2 Benchmark Strategies

We compare the performances of the bid prices obtained by the following five strategies.

• Sample path-based derivatives with randomized decision rule (SDR). This is the strategy that we develop in this paper but our implementation deviates from the previous discussion in two important aspects. First, since it is not realistic to assume that we can accept a fraction of an itinerary request, our implementation assumes that \( \theta(\cdot) \) characterizes the probability of accepting an itinerary request.
In particular, SDR computes the bid prices by solving problem (7). Letting \( \lambda^* \) be these bid prices, if there is enough capacity, then SDR accepts itinerary request \( t \) with probability \( \theta(r_t - \sum_{i \in \mathcal{L}} a_{it} \lambda^*_i) \). Otherwise, it rejects the itinerary request. In our computational experiments, we use 

\[
\theta(p) = \begin{cases} 
1 - a \exp\left(\frac{(1 - a)p}{b}\right) & \text{if } p \geq 0 \\
(1 - a) \exp\left(\frac{2a}{b}p\right) & \text{if } p < 0,
\end{cases}
\]  

(17)

with \( a \in (0, 1) \) and \( b > 0 \). Figure 2 plots the function in (17) and shows that this function looks like the step function as \( b \) approaches to zero. Therefore, the distinction between assuming that \( \theta(\cdot) \) characterizes a fraction or a probability diminishes as \( b \) approaches to zero. Our results are relatively insensitive to the choice of \( (a, b) \) and we use \( (a, b) = (1/2, 20/3) \) in our computational experiments. Nevertheless, we note that choosing \( b \) too small creates numerical difficulties.

Second, SDR recomputes the bid prices \( n \) times over the decision horizon by resolving problem (7) after every \( \tau/n \) itinerary requests. Given the remaining leg capacities just before making the decision for itinerary request \( t \), we compute the bid prices by solving the problem \( \min_{\lambda} \mathbb{E}\{R_t(x_t, \omega, \alpha, \lambda)\} \) and use these bid prices until we resolve problem (7). We use \( n \in \{1, 3, 6\} \) in our computational experiments.

We use the step size parameter \( \sigma^k = 20/(40 + k) \) in Algorithm 1 and terminate the algorithm after 20,000 iterations. The average CPU time required to solve problem (7) on a Pentium IV Desktop PC with 2.4 GHz CPU and 1 GB RAM is 34 seconds for the test problems with \( N = 6 \) and 168 seconds for the test problems with \( N = 12 \).

- Sample path-based derivatives with deterministic decision rule (SDD). In certain practical settings, it may not be realistic to use the randomized decision rule prescribed by SDR. As a remedy, SDD simply uses the bid prices obtained by solving problem (7) in the decision function in (1). In particular, SDD computes the bid prices by solving problem (7). Letting \( \lambda^* \) be these bid prices, if there is enough capacity and we have \( r_t \geq \sum_{i \in \mathcal{L}} a_{it} \lambda^*_i \), then SDD accepts itinerary request \( t \). Otherwise, it rejects the itinerary request. Similar to SDR, SDD recomputes the bid prices \( n \) times over the decision horizon.

We emphasize that the results in Section 3 are under the assumption that we can accept a fraction of an itinerary request. Therefore, SDR and SDD should be visualized only as practical extensions of the results in Section 3.

- Deterministic linear program (DLP). Using the notation at the beginning of Section 1 and noting that \( \sum_{t=1}^\tau 1(J_t = j) \) is the number of requests for itinerary \( j \) over the decision horizon, DLP computes the bid prices by solving the linear program

\[
\max \sum_{j \in \mathcal{J}} \tilde{r}_j z_j
\]

subject to 

\[
\sum_{j \in \mathcal{J}} \tilde{a}_{ij} z_j \leq x_{i1} \quad \text{for all } i \in \mathcal{L}
\]

\[
0 \leq z_j \leq \sum_{t=1}^\tau \mathbb{E}\{1(J_t = j)\} \quad \text{for all } j \in \mathcal{J};
\]

(18) (19) (20)

see Talluri and van Ryzin (2004). The decision variable \( z_j \) in the problem above is the number of
requests for itinerary $j$ that we plan to accept over the decision horizon. Constraints (19) ensure that the itinerary requests that we plan to accept do not violate the leg capacities, whereas constraints (20) ensure that the itinerary requests that we plan to accept do not exceed the expected numbers of the itinerary requests. Letting $\{\lambda^*_i : i \in L\}$ be the optimal values of the dual variables associated with constraints (19), if there is enough capacity and we have $r_t \geq \sum_{i \in L} a_{it} \lambda^*_i$, then DLP accepts itinerary request $t$. Otherwise, it rejects the itinerary request. It is also possible to show that the optimal objective value of problem (18)-(20) provides an upper bound on the total expected revenue obtained by the optimal policy. This information becomes useful when assessing the optimality gap of a benchmark strategy.

Similar to SDR, DLP recomputes the bid prices $n$ times over the decision horizon by resolving problem (18)-(20) after every $\tau/n$ itinerary requests. Given the remaining leg capacities just before making the decision for itinerary request $t$, we replace the right side of constraints (19) with $\{x_{it} : i \in L\}$ and the right side of constraints (20) with $\{\sum_{t'=t}^\tau \mathbb{E}\{1(J_{t'} = j)\} : j \in J\}$, and solve problem (18)-(20). We use the optimal values of the dual variables associated with constraints (19) as the bid prices until we resolve problem (18)-(20).

- Randomized linear program (RLP). DLP uses only the expected numbers of the itinerary requests and RLP tries to make up for this deficiency. The idea behind RLP is to replace the right side of constraints (20) with the samples of $\{\sum_{t=1}^\tau 1(J_t = j) : j \in J\}$. We generate $K$ samples of $\{J_t : t = 1, \ldots, \tau\}$ and denote these samples by $\{J^k_t : t = 1, \ldots, \tau, k\}$ for $k = 1, \ldots, K$. We replace the right side of constraints (20) with $\{\sum_{t=1}^\tau 1(J^k_t = j) : j \in J\}$ and solve problem (18)-(20). Letting $\{\lambda^*_{k,i} : i \in L\}$ be the optimal values of the dual variables associated with constraints (19), RLP uses $\{\sum_{k=1}^K \lambda^*_{k,i} / K : i \in L\}$ as the bid prices; see Talluri and van Ryzin (1999). We use $K = 25$ in our computational experiments.

- Finite differences on the deterministic linear program (FD). FD tries to improve on DLP by capturing the total opportunity cost of the leg capacities consumed by an itinerary request more accurately. Letting $L_1(x_1)$ be the optimal objective value of problem (18)-(20), we replace the right side of constraints (19) with $\{x_{i1} - \tilde{a}_{ij} : i \in L\}$ and solve problem (18)-(20) to obtain the optimal objective value $L^-_{j1}(x_1)$. If there is enough capacity and we have $r_t \geq L_1(x_1) - L^-_{j1}(x_1)$, then FD accepts itinerary request $t$. Otherwise, it rejects the itinerary request; see Bertsimas and Popescu (2003). Both RLP and FD recompute the bid prices $n$ times over the decision horizon by using an approach similar to the one used by DLP.

4.3 Computational Results

Our main computational results are summarized in three tables. Specifically, Tables 1, 2 and 3 respectively show the results for the cases where we recompute the bid prices once, three times and six times over the decision horizon. The first five columns in these tables respectively show the total expected revenues obtained by SDR, SDD, DLP, RLP and FD. We estimate these total expected revenues through simulation and use common random numbers when simulating the performances of the bid prices obtained by different strategies. The next four columns show the percent difference between the total expected revenues obtained by SDD and the other four strategies. SDD turns out to be one of the
better strategies and we use it as a reference point. The last four columns compare the performance of SDD with the performances of SDR, DLP, RLP and FD. In particular, the tenth column includes a “✓” if SDD performs better than SDR, a “×” if SDR performs better than SDD and a “◦” if there does not exist a statistically significant difference between the performances of SDD and SDR at 95% significance level. The interpretations of the eleventh, twelfth and thirteenth columns are similar but they respectively compare the performance of SDD with the performances of DLP, RLP and FD. Table 4 shows the optimal objective value of problem (18)-(20). As mentioned in the previous section, this is useful to get a feel for the optimality gap of different strategies.

Table 1 indicates that if we compute the bid prices only once at the beginning of the decision horizon, then SDD performs significantly better than DLP, RLP and FD. The performance gap is especially large when there is a large difference between the revenues associated with the high-fare and low-fare itineraries. Tables 2 and 3, on the other hand, indicate that the performance gap between SDD and the other three strategies gets smaller as we recompute the bid prices more frequently. For example, in Table 1, the total expected revenues obtained by SDD exceed those obtained by RLP by 10% on the average. If we recompute the bid prices three times over the decision horizon, then the total expected revenues obtained by SDD exceed those obtained by RLP by 3% on the average. If we recompute the bid prices six times over the decision horizon, then the same performance gap reduces to 1%. Nevertheless, we emphasize that such seemingly small performance gaps are quite significant in the network revenue management setting.

A general observation from the tables in this section is that SDD performs better than DLP, RLP and FD for a majority of the test problems. Specifically, SDD performs worse than RLP in only two cases and SDD performs worse than FD in only one case. SDD performs better than DLP in all of the cases. It is also interesting to note that the performances of DLP, RLP and FD improve significantly when we recompute the bid prices three or six times over the decision horizon. On the other hand, the performance of SDD is fairly satisfactory even when we compute the bid prices only once at the beginning of the decision horizon. It is possible to find test problems in Tables 1 and 2 for which the performance of SDD with $n = 1$ is better than the performance of DLP, RLP or FD with $n = 3$.

If we compute the bid prices only once at the beginning of the decision horizon, then SDR performs better than SDD. This is not surprising, since the results in Section 3 are under the assumption that we can accept a fraction of an itinerary request and SDR partially accommodates this assumption by using a randomized decision rule. However, it is surprising that if we recompute the bid prices three or six times over the decision horizon, then SDD performs noticeably better than SDR. This suggests that a choice between SDR and SDD should be made by considering the number of times that we recompute the bid prices and by comparing the performances of SDR and SDD in a specific problem context.

For our test problems, the performance of Algorithm 1 is relatively insensitive to the choice of the initial bid prices. In the computational experiments that we present in this section, we choose the initial bid prices as $\{\sum_{j \in J} \hat{a}_{ij}\hat{r}_j / \sum_{j \in J} \hat{a}_{ij} : i \in \mathcal{L}\}$. However, choosing the initial bid prices in a different manner does not yield drastically different results. For example, Figure 3 plots $E\{R_1(x_1, \omega, \alpha, \lambda^k)\}$ for test problem (12,1.6,8) as a function of the iteration counter $k$ in Algorithm 1. The three data
series correspond to the cases where we choose the initial bid prices as \( \{ \sum_{j \in J} \tilde{a}_{ij} \tilde{r}_j / \sum_{j \in J} \tilde{a}_{ij} : i \in \mathcal{L} \} \), as zero and as the bid prices obtained by the deterministic linear program. The figure indicates that the differences in the objective function values that we obtain after 10,000 iterations are less than 1%. Despite these encouraging empirical results, we emphasize the objective function of problem (7) is not concave and the performance of Algorithm 1 may potentially depend on the choice of the initial bid prices. Figure 3 also indicates that the performance of Algorithm 1 stabilizes after about 5,000 iterations. Nevertheless, to compensate for the lack of good stopping criteria for stochastic approximation methods and to be on the safe side, we terminate the algorithm after 20,000 iterations.

5 Conclusions

In this paper, we developed a convergent stochastic approximation method to compute bid prices in network revenue management problems. To facilitate the convergence proof, we worked with a smoothed version of the problem, which assumes that the leg capacities are continuous and we can accept a fraction of an itinerary request. SDR used the bid prices obtained by our stochastic approximation method through a randomized decision rule. Since such a randomized decision rule may not be realistic in certain practical settings, SDD used the bid prices obtained by our stochastic approximation method through the decision function in (1). Computational experiments demonstrated that the bid prices obtained by our stochastic approximation method are especially advantageous when there are multiple fare classes with large differences in the fares and the bid prices are not recomputed frequently.

There are several directions for further research. First, it is possible to use different sets of bid prices to make the decisions for different itinerary requests. Specifically, we can use the bid prices \( \{ \lambda_{it} : i \in \mathcal{L} \} \) to make the decision for itinerary request \( t \), in which case the cumulative revenue function in (6) becomes a function of \( \{ \lambda_{it} : i \in \mathcal{L}, \ t = 1, \ldots, \tau \} \). The difficulty with this approach is that the sample path-based derivative of the cumulative revenue function with respect to \( \lambda_{it} \) is zero when \( a_{it} = 0 \) and it is not always possible to obtain a meaningful sample path-based derivative with respect to all bid prices. Our preliminary computational experiments indicate that using different sets of bid prices to make the decisions for different itinerary requests does not provide a noticeable advantage over using the same set of bid prices to make the decisions for all itinerary requests. More work is needed to make this approach work. Second, as \( b \) approaches to zero, the function in (17) looks like the step function and the distinction between assuming that \( \theta(\cdot) \) characterizes a fraction or a probability diminishes. It is possible to visualize a version of Algorithm 1 where we decrease \( b \) at each iteration. It would be interesting to get a convergence result for this version as \( b \) approaches to zero and the number of iterations approaches to infinity. Third, the network revenue management problem with customer choice behavior is an emerging research area; see van Ryzin and Liu (2004). It is appealing to see whether a stochastic approximation method similar to ours can be useful in this setting.

A Appendix: Proposition 4.1 in Bertsekas and Tsitsiklis (1996)

For a function \( f(\cdot) : \mathbb{R}^n \to \mathbb{R} \), we consider the algorithm

\[
\lambda^{k+1} = \lambda^k + \sigma^k s^k
\]
to solve the problem \( \max_\lambda f(\lambda) \), where \( \{\sigma^k\}_k \) is a sequence of step size parameters and \( \{s^k\}_k \) is a sequence of step directions. We let \( \mathcal{F}^k \) be the filtration generated by the random variables \( \{\lambda^1, s^1, \ldots, s^{k-1}\} \) in this algorithm and assume that the following statements hold for all \( \lambda, \gamma \in \mathbb{R}^n \).

(B.0) We have \( f(\lambda) \geq 0 \).
(B.1) We have \( \mathbb{E}\{s^k | \mathcal{F}^k\} = \nabla f(\lambda^k) \) w.p.1 for all \( k = 1, 2, \ldots \).
(B.2) There exists a finite scalar \( M_s \) such that we have \( \|s^k\| \leq M_s \) w.p.1 for all \( k = 1, 2, \ldots \).
(B.3) There exists a finite scalar \( L_f \) such that we have \( \|\nabla f(\lambda) - \nabla f(\gamma)\| \leq L_f \|\lambda - \gamma\| \).

In this case, the next convergence result is from Proposition 4.1 in Bertsekas and Tsitsiklis (1996).

**Proposition 7** Assume that the sequence of step size parameters \( \{\sigma^k\}_k \) are \( \mathcal{F}^k \)-measurable and satisfy \( \sigma^k \geq 0 \) for all \( k = 1, 2, \ldots, \sum_{k=1}^{\infty} \sigma^k = \infty \) and \( \sum_{k=1}^{\infty} |\sigma^k|^2 < \infty \) w.p.1. If the sequence \( \{\lambda^k\}_k \) is generated by the algorithm above and (B.0)-(B.3) hold, then we have \( \lim_{k \to \infty} \nabla f(\lambda^k) = 0 \) w.p.1 and every limit point \( \lambda^* \) of the sequence \( \{\lambda^k\}_k \) satisfies \( \nabla f(\lambda^*) = 0 \) w.p.1.

**B Appendix: Lemma 6.3.1 in Glasserman (1994)**

For a probability space \((\Omega, \mathcal{F}, \mathbb{P})\) and a function \( f(\cdot, \cdot) : \mathbb{R}^n \times \Omega \to \mathbb{R} \), we assume that the following statements hold for all \( \lambda, \gamma \in \mathbb{R}^n \).

(C.1) The function \( f(\cdot, \omega) \) is differentiable at \( \lambda \) for \( \mathbb{P} \)-almost all values of \( \omega \).
(C.2) There exists a finite scalar \( L_f \) such that we have \( \|f(\lambda, \omega) - f(\gamma, \omega)\| \leq L_f \|\lambda - \gamma\| \) for \( \mathbb{P} \)-almost all values of \( \omega \).

In this case, the next result is from Lemma 6.3.1 in Glasserman (1994).

**Lemma 8** Assume that there exists a finite scalar \( M_f \) that satisfies \( \mathbb{E}\{|f(\lambda, \omega)|\} \leq M_f \) for all \( \lambda \in \mathbb{R}^n \). In this case, if (C.1) and (C.2) hold, then \( \nabla \mathbb{E}\{f(\lambda, \omega)\} \) exists and we have \( \nabla \mathbb{E}\{f(\lambda, \omega)\} = \mathbb{E}\{\nabla f(\lambda, \omega)\} \) for all \( \lambda \in \mathbb{R}^n \).

**C Appendix: Proofs of the Results Omitted in the Paper**

The proof of Lemma 4 uses Lemma 10 below. The next result is useful when showing Lemma 10.

**Lemma 9** If \( \|x_t - z_t\| \leq M \|\lambda - \gamma\| \) for some \( M \in \mathbb{R}_+ \), then we have \( \|u_t(x_t, \omega, \alpha, \lambda) - u_t(z_t, \omega, \alpha, \gamma)\| \leq (L_u + M) \|\lambda - \gamma\| \) w.p.1 for a finite scalar \( L_u \).

**Proof** We consider four cases.

**Case 1.** We assume that \( \theta(r_t - \sum_{j \in \mathcal{L}} a_j \gamma_j) \leq \min_{j \in \mathcal{L}_+} \{x_{j^*} + \alpha_j^* / a_j^*\} \) and \( \theta(r_t - \sum_{j \in \mathcal{L}} a_j \gamma_j) \leq \min_{j \in \mathcal{L}_+} \{z_{j^*} + \alpha_j^* / a_j^*\} \). Using (5), we have \( \|u_t(x_t, \omega, \alpha, \lambda) - u_t(z_t, \omega, \alpha, \gamma)\| = \|\theta(r_t - \sum_{j \in \mathcal{L}} a_j \gamma_j) - \theta(r_t - \sum_{j \in \mathcal{L}} a_j \gamma_j)\| \leq L_{\theta} \|\sum_{j \in \mathcal{L}} a_j (\lambda_j - \gamma_j)\| \leq B_a \|L\| \lambda - \gamma\|.$
Case 2. We assume that \( \theta(r_t - \sum_{j \in L} a_{jt} \lambda_j) \leq \min_{j \in L^+} \{ [x_{jt} + \alpha_{jt}] / a_{jt} \} \) and \( \theta(r_t - \sum_{j \in L} a_{jt} \gamma_j) > \min_{j \in L^+} \{ [z_{jt} + \alpha_{jt}] / a_{jt} \} \). Using (5), we have \( |u_t(x_t, \omega, \alpha, \lambda) - u_t(z_t, \omega, \alpha, \gamma)| = |\theta(r_t - \sum_{j \in L} a_{jt} \lambda_j) - \theta(r_t - \sum_{j \in L} a_{jt} \gamma_j)| \) and we consider two subcases.

Case 2.a. We assume that \( \theta(r_t - \sum_{j \in L} a_{jt} \lambda_j) \geq \theta(r_t - \sum_{j \in L} a_{jt} \gamma_j) \), in which case we have \( \min_{j \in L^+} \{ [z_{jt} + \alpha_{jt}] / a_{jt} \} < \theta(r_t - \sum_{j \in L} a_{jt} \lambda_j) \leq \theta(r_t - \sum_{j \in L} a_{jt} \gamma_j) \). Since \( z_{jt} - M \| \lambda - \gamma \| \leq x_{jt} \leq z_{jt} + M \| \lambda - \gamma \| \) for all \( j \in L \) and \( M \| \lambda - \gamma \| / a_{jt} \leq M \| \lambda - \gamma \| \) for all \( j \in L^+ \), we obtain

\[
|u_t(x_t, \omega, \alpha, \lambda) - u_t(z_t, \omega, \alpha, \gamma)| \leq \min_{j \in L^+} \{ [x_{jt} + \alpha_{jt}] / a_{jt} \} - \min_{j \in L^+} \{ [z_{jt} + \alpha_{jt}] / a_{jt} \} \\
\leq \min_{j \in L^+} \{ [z_{jt} + M \| \lambda - \gamma \| + \alpha_{jt}] / a_{jt} \} - \min_{j \in L^+} \{ [z_{jt} + \alpha_{jt}] / a_{jt} \} \\
\leq M \| \lambda - \gamma \|.
\]

Case 2.b. We assume that \( \theta(r_t - \sum_{j \in L} a_{jt} \lambda_j) < \theta(r_t - \sum_{j \in L} a_{jt} \gamma_j) \) and consider two (sub)subcases.

Case 2.b.i. We assume that \( \theta(r_t - \sum_{j \in L} a_{jt} \lambda_j) \geq \min_{j \in L^+} \{ [z_{jt} + \alpha_{jt}] / a_{jt} \} \), in which case we have \( \min_{j \in L^+} \{ [z_{jt} + \alpha_{jt}] / a_{jt} \} \leq \theta(r_t - \sum_{j \in L} a_{jt} \lambda_j) \). Therefore, we obtain

\[
|u_t(x_t, \omega, \alpha, \lambda) - u_t(z_t, \omega, \alpha, \gamma)| \leq \min_{j \in L^+} \{ [x_{jt} + \alpha_{jt}] / a_{jt} \} - \min_{j \in L^+} \{ [z_{jt} + \alpha_{jt}] / a_{jt} \} \leq M \| \lambda - \gamma \|
\]

where the second inequality follows from the same argument in Case 2.a.

Case 2.b.ii. Assume that \( \theta(r_t - \sum_{j \in L} a_{jt} \lambda_j) < \min_{j \in L^+} \{ [z_{jt} + \alpha_{jt}] / a_{jt} \} \), in which case we have \( \theta(r_t - \sum_{j \in L} a_{jt} \gamma_j) = \theta(r_t - \sum_{j \in L} a_{jt} \lambda_j) \). Therefore, we obtain

\[
|u_t(x_t, \omega, \alpha, \lambda) - u_t(z_t, \omega, \alpha, \gamma)| \leq \theta(r_t - \sum_{j \in L} a_{jt} \gamma_j) - \theta(r_t - \sum_{j \in L} a_{jt} \lambda_j) \leq B_a |L| \| \lambda - \gamma \|.
\]

The other cases that we do not cover above can be handled in a similar manner. If we combine all cases, then it is easy to see that letting \( L_u = B_a |L| \| \lambda - \gamma \| \) suffices.

\[
\square
\]

Lemma 10 We have \( \| x_t^\lambda - x_t^\gamma \| \leq L_X \| \lambda - \gamma \| \) w.p.1 for a finite scalar \( L_X \).

Proof All statements in the proof are in w.p.1 sense. We show by induction that \( \| x_t^\lambda - x_t^\gamma \| \leq \left[ 2(1 + B_a) \left( L_u + \ldots + 2^{t-1}(1 + B_a)^{t-1} L_u \right) \right] \| \lambda - \gamma \| \) for all \( t = 2, \ldots, \tau \), in which case the result follows by letting \( L_X = 2(1 + B_a) \left( L_u + \ldots + 2^{t-1}(1 + B_a)^{t-1} L_u \right) \) and noting that \( x_t^\lambda = x_t^\gamma = x_1 \). Assuming that the result holds for itinerary request \( t \) and using (13), we have

\[
\| x_{t+1}^\lambda - x_{t+1}^\gamma \| \leq \| x_t^\lambda - x_t^\gamma \| + B_a |u_t(x_t^\lambda, \omega, \alpha, \lambda) - u_t(x_t^\gamma, \omega, \alpha, \gamma)| \\
\leq (1 + B_a) \| x_t^\lambda - x_t^\gamma \| + (1 + B_a) |u_t(x_t^\lambda, \omega, \alpha, \lambda) - u_t(x_t^\gamma, \omega, \alpha, \gamma)| \\
\leq (1 + B_a) \left[ 2(1 + B_a) \left( L_u + \ldots + 2^{t-1}(1 + B_a)^{t-1} L_u \right) \| \lambda - \gamma \| \right] + (1 + B_a) \left[ L_u + 2(1 + B_a) L_u + \ldots + 2^{t-1}(1 + B_a)^{t-1} L_u \right] \| \lambda - \gamma \| \\
\leq \left[ 2(1 + B_a) L_u + 2^t (1 + B_a)^2 L_u + \ldots + 2^t (1 + B_a)^t L_u \right] \| \lambda - \gamma \|,
\]

where the third inequality follows from the induction hypothesis and Lemma 9. Therefore, the result holds for itinerary request \( t + 1 \). We complete the induction argument by noting that

\[
\| x^\lambda_2 - x^\gamma_2 \| \leq \| x^\lambda_1 - x^\gamma_1 \| + B_a |u_1(x^\lambda_1, \omega, \alpha, \lambda) - u_1(x^\gamma_1, \omega, \alpha, \gamma)| \leq B_a L_u \| \lambda - \gamma \| \leq 2(1 + B_a) L_u \| \lambda - \gamma \|,
\]
where we use Lemma 9 and the fact that \( \| x^λ_1 - x^λ_2 \| \leq 0 \| \lambda - \gamma \|. \)

The next result is useful when showing Lemma 5.

**Lemma 11** We have \( \mathbb{E}\{ | \partial^X_i u_t(x^λ_i, \omega, \alpha, \lambda) - \partial^X_i u_t(x^γ_i, \omega, \alpha, \gamma)| \} \leq L^X_i \| \lambda - \gamma \| \) for a finite scalar \( L^X_i \).

**Proof** By (11), \( | \partial^X_i u_t(x^λ_i, \omega, \alpha, \lambda) - \partial^X_i u_t(x^γ_i, \omega, \alpha, \gamma)| \) is equal to \( 1/a_{it} \) for the following four cases and is equal to zero otherwise.

**Case 1.** \( i = \arg\min_{j \in \mathcal{L}_t^+} \{ [x^λ_{jt} + \alpha_{jt}] / a_{jt} \}, [x^λ_{it} + \alpha_{it}] / a_{it} \leq \theta (r_t - \sum_{j \in \mathcal{L}} a_{jt} \lambda_j) \),

\[ i = \arg\min_{j \in \mathcal{L}_t^+} \{ [x^λ_{jt} + \alpha_{jt}] / a_{jt} \}, [x^λ_{it} + \alpha_{it}] / a_{it} > \theta (r_t - \sum_{j \in \mathcal{L}} a_{jt} \gamma_j) \).

**Case 2.** \( i = \arg\min_{j \in \mathcal{L}_t^+} \{ [x^γ_{jt} + \alpha_{jt}] / a_{jt} \}, [x^γ_{it} + \alpha_{it}] / a_{it} < \theta (r_t - \sum_{j \in \mathcal{L}} a_{jt} \lambda_j) \),

\[ i = \arg\min_{j \in \mathcal{L}_t^+} \{ [x^γ_{jt} + \alpha_{jt}] / a_{jt} \}, [x^γ_{it} + \alpha_{it}] / a_{it} > \theta (r_t - \sum_{j \in \mathcal{L}} a_{jt} \gamma_j) \).

**Case 3.** \( i = \arg\min_{j \in \mathcal{L}_t^+} \{ [x^γ_{jt} + \alpha_{jt}] / a_{jt} \}, [x^λ_{it} + \alpha_{it}] / a_{it} \leq \theta (r_t - \sum_{j \in \mathcal{L}} a_{jt} \lambda_j) \),

\[ i \neq \arg\min_{j \in \mathcal{L}_t^+} \{ [x^γ_{jt} + \alpha_{jt}] / a_{jt} \}. \]

**Case 4.** \( i \neq \arg\min_{j \in \mathcal{L}_t^+} \{ [x^γ_{jt} + \alpha_{jt}] / a_{jt} \}, i = \arg\min_{j \in \mathcal{L}_t^+} \{ [x^λ_{jt} + \alpha_{jt}] / a_{jt} \}, [x^γ_{it} + \alpha_{it}] / a_{it} < \theta (r_t - \sum_{j \in \mathcal{L}} a_{jt} \lambda_j) \).

Since we have either \( i = \arg\min_{j \in \mathcal{L}_t^+} \{ [x^λ_{jt} + \alpha_{jt}] / a_{jt} \} \) or \( i = \arg\min_{j \in \mathcal{L}_t^+} \{ [x^γ_{jt} + \alpha_{jt}] / a_{jt} \} \) for these four cases and \( a_{it} \geq 1 \) for all \( i \in \mathcal{L}_t^+ \), we also have \( | \partial^X_i u_t(x^λ_i, \omega, \alpha, \lambda) - \partial^X_i u_t(x^γ_i, \omega, \alpha, \gamma)| \leq 1. \)

We obtain a bound on the probability of the first case by noting that

\[ \mathbb{P}\left\{ i = \arg\min_{j \in \mathcal{L}_t^+} \{ [x^λ_{jt} + \alpha_{jt}] / a_{jt} \}, [x^λ_{it} + \alpha_{it}] / a_{it} \leq \theta (r_t - \sum_{j \in \mathcal{L}} a_{jt} \lambda_j) \), \right. \]

\[ = \mathbb{P}\left\{ a_{it} \theta (r_t - \sum_{j \in \mathcal{L}} a_{jt} \lambda_j) - x^λ_{it} \leq \alpha_{it} \theta (r_t - \sum_{j \in \mathcal{L}} a_{jt} \lambda_j) - x^λ_{it} \right\}. \]

By (14), the probability on right side above is bounded by \( [B^2_\theta | \mathcal{L} | L_\theta + L_X] \| \lambda - \gamma \| / \epsilon \). The same bound applies to the probability of the second case. Using \( \mathbb{P}_{\omega}\{ \cdot \} \) to denote probability conditional on the filtration generated by \( \omega \), we obtain a bound on the probability of the third case by noting that

\[ \mathbb{P}_{\omega}\left\{ i = \arg\min_{j \in \mathcal{L}_t^+} \{ [x^λ_{jt} + \alpha_{jt}] / a_{jt} \}, [x^γ_{it} + \alpha_{it}] / a_{it} \leq \theta (r_t - \sum_{j \in \mathcal{L}} a_{jt} \lambda_j) \right. \]

\[ = \mathbb{P}_{\omega}\left\{ i = \arg\min_{j \in \mathcal{L}_t^+} \{ [x^γ_{jt} + \alpha_{jt}] / a_{jt} \}, i \neq \arg\min_{j \in \mathcal{L}_t^+} \{ [x^λ_{jt} + \alpha_{jt}] / a_{jt} \} \right\} \]

\[ = \sum_{i' \in \mathcal{L}_t^+ \setminus \{i\}} \mathbb{P}_{\omega}\left\{ [x^λ_{jt} + \alpha_{jt}] / a_{jt} < [x^γ_{it} + \alpha_{it}] / a_{it}, \right. \]

\[ \left. [x^γ_{jt} + \alpha_{jt}] / a_{jt} < [x^γ_{it} + \alpha_{it}] / a_{it} \right\} \]

\[ = \sum_{i' \in \mathcal{L}_t^+ \setminus \{i\}} \mathbb{P}_{\omega}\left\{ \frac{x^γ_{jt}}{a_{jt}} - \frac{x^γ_{it}}{a_{it}} < \frac{x^λ_{jt}}{a_{jt}} - \frac{x^λ_{it}}{a_{it}} \right\}. \]
Using the fact that the random variables \( \alpha_{it} \) and \( \alpha_{i't} \) are uniformly distributed over the interval \([0, \epsilon]\), and they are independent of each other and \( \omega \), a straightforward computation shows that

\[
\mathbb{P}_\omega \left\{ \frac{\alpha_{it}}{\alpha_{i't}} - \frac{\alpha_{i't}}{a_{i't}} \leq p \right\} = \begin{cases} 
0 & \text{if } p \leq -\frac{\epsilon}{a_{i't}} \\
\frac{a_{it}}{2 a_{i't}} + \frac{a_{it} p}{\epsilon} + \frac{a_{i't}}{2 a_{i't}} \left[ \frac{a_{i't}}{\epsilon} \right]^2 & \text{if } -\frac{\epsilon}{a_{i't}} < p \leq 0 \text{ and } \frac{a_{it}}{a_{i't}} + \frac{a_{it} p}{\epsilon} \leq 1 \\
1 + \frac{1}{2 a_{i't}} \left[ \frac{a_{i't}}{\epsilon} \right]^2 - \frac{a_{i't}}{2 a_{i't}} & \text{if } -\frac{\epsilon}{a_{i't}} < p \leq 0 \text{ and } \frac{a_{it}}{a_{i't}} + \frac{a_{it} p}{\epsilon} > 1 \\
1 - \frac{a_{i't}}{2 a_{i't}} \left[ 1 - \frac{a_{it} p}{\epsilon} \right]^2 & \text{if } 0 < p \leq \frac{\epsilon}{a_{i't}} \text{ and } \frac{a_{it}}{a_{i't}} + \frac{a_{it} p}{\epsilon} \leq 1 \\
1 & \text{if } 0 < p \leq \frac{\epsilon}{a_{i't}} \text{ and } \frac{a_{it}}{a_{i't}} + \frac{a_{it} p}{\epsilon} > 1 \\
\end{cases}
\]

Using this expression, it is easy to check that the cumulative distribution function of \( [\alpha_{it}/a_{it}] - [\alpha_{i't}/a_{i't}] \) conditional on \( \omega \) is Lipschitz with modulus \( B_a/\epsilon \) for all \( i, i' \in L_t^+ \). Therefore, we have

\[
\mathbb{P}_\omega \left\{ q \leq \frac{\alpha_{it}}{a_{it}} - \frac{\alpha_{i't}}{a_{i't}} \leq p \right\} \leq \frac{B_a}{\epsilon} |p-q|
\]

w.p.1 for all \( i, i' \in L_t^+ \). On the other hand, since we have \( a_{it} \geq 1 \) for all \( i \in L_t^+ \), Lemma 10 implies that

\[
\left| \left[ \frac{x_{i't}^\lambda}{a_{i't}} - \frac{x_{i't}^\lambda}{a_{it}} \right] - \left[ \frac{x_{i't}^\gamma}{a_{i't}} - \frac{x_{i't}^\gamma}{a_{it}} \right] \right| \leq \left| \left| x_{i't}^\lambda - x_{i't}^\lambda \right| + \left| x_{i't}^\gamma - x_{i't}^\gamma \right| \right| \leq 2 L \| \lambda - \gamma \|
\]

w.p.1 for all \( i, i' \in L_t^+ \). By the last two inequalities, we obtain

\[
\mathbb{P}_\omega \left\{ \frac{x_{i't}^\gamma}{a_{i't}} - \frac{x_{i't}^\gamma}{a_{it}} < \frac{x_{i't}^\lambda}{a_{i't}} - \frac{x_{i't}^\lambda}{a_{it}} \right\} \leq \frac{2 B_a}{\epsilon} L \| \lambda - \gamma \| \| \lambda - \gamma \| / \epsilon + 4 B_a |\mathcal{L}| L_X \left\| \lambda - \gamma \right\| / \epsilon \text{ and the result follows.} \quad (22)
\]

We are now ready to show Lemma 5. All statements in the proof are in w.p.1 sense. Using (13), we write (10) as

\[
\partial_t^X R_t(x_t^\lambda, \omega, \alpha, \lambda) = r_t \partial_t^X u_t(x_t^\lambda, \omega, \alpha, \lambda) + \sum_{j \in \mathcal{L}} \left[ 1(j = i) - a_{jt} \partial_t^X u_t(x_t^\lambda, \omega, \alpha, \lambda) \right] \partial_j^X R_{t+1}(x_{t+1}^\lambda, \omega, \alpha, \lambda). \quad (23)
\]

Since \( a_{it} \geq 1 \) for all \( i \in L_t^+ \), we have \( \left| \partial_t^X u_t(x_t^\lambda, \omega, \alpha, \lambda) \right| \leq 1 \) for all \( i \in \mathcal{L} \) by (11) and we can use an argument similar to the one in (16) to obtain

\[
\left| \partial_t^X u_t(x_t^\lambda, \omega, \alpha, \lambda) \partial_t^X R_{t+1}(x_{t+1}^\lambda, \omega, \alpha, \lambda) - \partial_t^X u_t(x_t^\gamma, \omega, \alpha, \gamma) \partial_t^X R_{t+1}(x_{t+1}^\gamma, \omega, \alpha, \gamma) \right| \\
\leq B_R^X \left| \partial_t^X u_t(x_t^\lambda, \omega, \alpha, \lambda) - \partial_t^X u_t(x_t^\gamma, \omega, \alpha, \gamma) \right| + | \partial_t^X R_{t+1}(x_{t+1}^\lambda, \omega, \alpha, \lambda) - \partial_t^X R_{t+1}(x_{t+1}^\gamma, \omega, \alpha, \gamma) |,
\]
where $B_R^X$ is as in the proof of Proposition 3. Therefore, (23) and the inequality above imply that

$$
\mathbb{E}\{ \partial^X_t R_t(x^\lambda_t, \omega, \alpha, \lambda) - \partial^X_t R_t(x^\gamma_t, \omega, \alpha, \gamma) \} \\
\leq B_r \mathbb{E}\{ \partial^X_t u_t(x^\lambda_t, \omega, \alpha, \lambda) - \partial^X_t u_t(x^\gamma_t, \omega, \alpha, \gamma) \} \\
+ \sum_{j \in \mathcal{L}} \mathbb{E}\{ \partial^X_t R_{t+1}(x^\lambda_{t+1}, \omega, \alpha, \lambda) - \partial^X_t R_{t+1}(x^\gamma_{t+1}, \omega, \alpha, \gamma) \} \\
+ \sum_{j \in \mathcal{L}} B_a B_R^X \mathbb{E}\{ \partial^X_t u_t(x^\lambda_t, \omega, \alpha, \lambda) - \partial^X_t u_t(x^\gamma_t, \omega, \alpha, \gamma) \} \\
+ \sum_{j \in \mathcal{L}} B_a \mathbb{E}\{ \partial^X_t R_{t+1}(x^\lambda_{t+1}, \omega, \alpha, \lambda) - \partial^X_t R_{t+1}(x^\gamma_{t+1}, \omega, \alpha, \gamma) \},
$$
in which case Lemma 11 implies that

$$
\mathbb{E}\{ \partial^X_t R_t(x^\lambda_t, \omega, \alpha, \lambda) - \partial^X_t R_t(x^\gamma_t, \omega, \alpha, \gamma) \} \\
\leq B_r L^X_a \| \lambda - \gamma \| + B_a B_R^X |\mathcal{L}| L^X_a \| \lambda - \gamma \| \\
\sum_{j \in \mathcal{L}} (1 + B_a) \mathbb{E}\{ \partial^X_t R_{t+1}(x^\lambda_{t+1}, \omega, \alpha, \lambda) - \partial^X_t R_{t+1}(x^\gamma_{t+1}, \omega, \alpha, \gamma) \}. \tag{24}
$$

Letting $M = B_r L^X_a + B_a B_R^X |\mathcal{L}|$, we now use (24) to show by induction that $\mathbb{E}\{ \partial^X_t R_t(x^\lambda_t, \omega, \alpha, \lambda) - \partial^X_t R_t(x^\gamma_t, \omega, \alpha, \gamma) \} \leq [M + (1 + B_a) |\mathcal{L}| M + \ldots + (1 + B_a)^{\tau - t} |\mathcal{L}|^{\tau - t - 1} M] \| \lambda - \gamma \|$ for all $i \in \mathcal{L}$, $t = 1, \ldots, \tau$, in which case the result follows by letting $L_a^{X_i} = M + (1 + B_a) |\mathcal{L}| M + \ldots + (1 + B_a)^{\tau - 1} |\mathcal{L}|^{\tau - 1} M$. Assuming that the result holds for itinerary request $t + 1$ and using (24), we have

$$
\mathbb{E}\{ \partial^X_t R_t(x^\lambda_t, \omega, \alpha, \lambda) - \partial^X_t R_t(x^\gamma_t, \omega, \alpha, \gamma) \} \\
\leq M \| \lambda - \gamma \| + (1 + B_a) |\mathcal{L}| [M + (1 + B_a) |\mathcal{L}| M + \ldots + (1 + B_a)^{\tau - 1} |\mathcal{L}|^{\tau - 1} M] \| \lambda - \gamma \| \\
= [M + (1 + B_a) |\mathcal{L}| M + \ldots + (1 + B_a)^{\tau - l} |\mathcal{L}|^{\tau - l - 1} M] \| \lambda - \gamma \|.
$$

We complete the induction argument by noting that $\mathbb{E}\{ \| \partial^X_t R_t(x^\lambda_t, \omega, \alpha, \lambda) - \partial^X_t R_t(x^\gamma_t, \omega, \alpha, \gamma) \| \} \leq M \| \lambda - \gamma \|$ by (24).

\begin{center}
\textbf{REFERENCES}
\end{center}


Figure 1: The structure of the airline network for the case where $N = 6$.

Figure 2: The function in (17) for different values of $b$.

Figure 3: The trajectory of Algorithm 1 for test problem $(12, 1.6, 8)$. We choose the initial bid prices (a) as $\left\{ \sum_{j \in J} \tilde{a}_{ij} \tilde{r}_j / \sum_{j \in J} \tilde{a}_{ij} : i \in L \right\}$, (b) as zero and (c) as the bid prices obtained by the deterministic linear program.
Table 1: Computational results for the case where \( n = 1 \).

<table>
<thead>
<tr>
<th>prob.</th>
<th>total expected revenue obtained by</th>
<th>% diff. btwn. SDD and SDR</th>
<th>sig. diff. btwn. SDD and SDR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SDR</td>
<td>SDD</td>
<td>DLP</td>
</tr>
<tr>
<td>(6,1.0,2)</td>
<td>16,956</td>
<td>16,948</td>
<td>16,538</td>
</tr>
<tr>
<td>(6,1.0,4)</td>
<td>24,052</td>
<td>23,552</td>
<td>22,295</td>
</tr>
<tr>
<td>(6,1.0,8)</td>
<td>38,945</td>
<td>38,669</td>
<td>33,809</td>
</tr>
<tr>
<td>(6,1.2,2)</td>
<td>14,941</td>
<td>14,786</td>
<td>14,304</td>
</tr>
<tr>
<td>(6,1.2,4)</td>
<td>21,904</td>
<td>21,055</td>
<td>18,563</td>
</tr>
<tr>
<td>(6,1.2,8)</td>
<td>36,023</td>
<td>34,803</td>
<td>27,110</td>
</tr>
<tr>
<td>(6,1.6,2)</td>
<td>12,061</td>
<td>12,006</td>
<td>11,313</td>
</tr>
<tr>
<td>(6,1.6,4)</td>
<td>18,889</td>
<td>18,687</td>
<td>14,867</td>
</tr>
<tr>
<td>(6,1.6,8)</td>
<td>33,221</td>
<td>32,677</td>
<td>21,976</td>
</tr>
<tr>
<td>(12,1.0,2)</td>
<td>30,611</td>
<td>30,491</td>
<td>29,725</td>
</tr>
<tr>
<td>(12,1.0,4)</td>
<td>43,908</td>
<td>43,740</td>
<td>39,936</td>
</tr>
<tr>
<td>(12,1.0,8)</td>
<td>71,718</td>
<td>71,370</td>
<td>60,638</td>
</tr>
<tr>
<td>(12,1.2,2)</td>
<td>27,390</td>
<td>27,697</td>
<td>26,481</td>
</tr>
<tr>
<td>(12,1.2,4)</td>
<td>40,436</td>
<td>40,494</td>
<td>35,106</td>
</tr>
<tr>
<td>(12,1.2,8)</td>
<td>67,189</td>
<td>66,602</td>
<td>52,353</td>
</tr>
<tr>
<td>(12,1.6,2)</td>
<td>22,447</td>
<td>22,905</td>
<td>21,993</td>
</tr>
<tr>
<td>(12,1.6,4)</td>
<td>35,084</td>
<td>34,577</td>
<td>30,214</td>
</tr>
<tr>
<td>(12,1.6,8)</td>
<td>61,809</td>
<td>60,147</td>
<td>46,363</td>
</tr>
</tbody>
</table>

Table 2: Computational results for the case where \( n = 3 \).

<table>
<thead>
<tr>
<th>prob.</th>
<th>total expected revenue obtained by</th>
<th>% diff. btwn. SDD and SDR</th>
<th>sig. diff. btwn. SDD and SDR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SDR</td>
<td>SDD</td>
<td>DLP</td>
</tr>
<tr>
<td>(6,1.0,2)</td>
<td>17,185</td>
<td>17,343</td>
<td>17,082</td>
</tr>
<tr>
<td>(6,1.0,4)</td>
<td>24,570</td>
<td>24,715</td>
<td>23,766</td>
</tr>
<tr>
<td>(6,1.0,8)</td>
<td>39,301</td>
<td>39,282</td>
<td>37,133</td>
</tr>
<tr>
<td>(6,1.2,2)</td>
<td>15,312</td>
<td>15,517</td>
<td>15,209</td>
</tr>
<tr>
<td>(6,1.2,4)</td>
<td>22,556</td>
<td>22,711</td>
<td>21,170</td>
</tr>
<tr>
<td>(6,1.2,8)</td>
<td>35,585</td>
<td>35,603</td>
<td>33,211</td>
</tr>
<tr>
<td>(6,1.6,2)</td>
<td>12,495</td>
<td>12,667</td>
<td>12,266</td>
</tr>
<tr>
<td>(6,1.6,4)</td>
<td>19,673</td>
<td>19,712</td>
<td>17,569</td>
</tr>
<tr>
<td>(6,1.6,8)</td>
<td>33,261</td>
<td>32,974</td>
<td>28,206</td>
</tr>
<tr>
<td>(12,1.0,2)</td>
<td>31,114</td>
<td>31,508</td>
<td>30,930</td>
</tr>
<tr>
<td>(12,1.0,4)</td>
<td>44,647</td>
<td>45,187</td>
<td>43,291</td>
</tr>
<tr>
<td>(12,1.0,8)</td>
<td>72,586</td>
<td>73,041</td>
<td>68,053</td>
</tr>
<tr>
<td>(12,1.2,2)</td>
<td>28,017</td>
<td>28,510</td>
<td>27,971</td>
</tr>
<tr>
<td>(12,1.2,4)</td>
<td>41,261</td>
<td>41,754</td>
<td>39,279</td>
</tr>
<tr>
<td>(12,1.2,8)</td>
<td>68,775</td>
<td>69,199</td>
<td>62,101</td>
</tr>
<tr>
<td>(12,1.6,2)</td>
<td>23,073</td>
<td>23,658</td>
<td>23,217</td>
</tr>
<tr>
<td>(12,1.6,4)</td>
<td>35,948</td>
<td>36,290</td>
<td>34,008</td>
</tr>
<tr>
<td>(12,1.6,8)</td>
<td>63,199</td>
<td>63,387</td>
<td>55,894</td>
</tr>
<tr>
<td>prob.</td>
<td>total expected revenue obtained by</td>
<td>% diff. btwn. SDD and</td>
<td>sig. diff. btwn. SDD and</td>
</tr>
<tr>
<td>-------</td>
<td>----------------------------------</td>
<td>-----------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td></td>
<td>SDR</td>
<td>SDD</td>
<td>DLP</td>
</tr>
<tr>
<td>(6,1.0,2)</td>
<td>17,277</td>
<td>17,442</td>
<td>17,232</td>
</tr>
<tr>
<td>(6,1.0,4)</td>
<td>24,671</td>
<td>24,923</td>
<td>24,132</td>
</tr>
<tr>
<td>(6,1.0,8)</td>
<td>39,158</td>
<td>39,159</td>
<td>37,933</td>
</tr>
<tr>
<td>(6,1.2,2)</td>
<td>15,440</td>
<td>15,683</td>
<td>15,433</td>
</tr>
<tr>
<td>(6,1.2,4)</td>
<td>22,711</td>
<td>23,000</td>
<td>21,807</td>
</tr>
<tr>
<td>(6,1.2,8)</td>
<td>36,504</td>
<td>36,415</td>
<td>34,646</td>
</tr>
<tr>
<td>(6,1.6,2)</td>
<td>12,597</td>
<td>12,824</td>
<td>12,571</td>
</tr>
<tr>
<td>(6,1.6,4)</td>
<td>19,844</td>
<td>20,031</td>
<td>18,519</td>
</tr>
<tr>
<td>(6,1.6,8)</td>
<td>33,023</td>
<td>33,267</td>
<td>30,503</td>
</tr>
<tr>
<td>(12,1.0,2)</td>
<td>31,240</td>
<td>31,666</td>
<td>31,231</td>
</tr>
<tr>
<td>(12,1.0,4)</td>
<td>44,856</td>
<td>45,351</td>
<td>43,928</td>
</tr>
<tr>
<td>(12,1.0,8)</td>
<td>72,692</td>
<td>73,265</td>
<td>69,435</td>
</tr>
<tr>
<td>(12,1.2,2)</td>
<td>28,204</td>
<td>28,733</td>
<td>28,294</td>
</tr>
<tr>
<td>(12,1.2,4)</td>
<td>41,483</td>
<td>42,001</td>
<td>40,108</td>
</tr>
<tr>
<td>(12,1.2,8)</td>
<td>69,006</td>
<td>69,593</td>
<td>64,069</td>
</tr>
<tr>
<td>(12,1.6,2)</td>
<td>23,303</td>
<td>23,897</td>
<td>23,541</td>
</tr>
<tr>
<td>(12,1.6,4)</td>
<td>36,236</td>
<td>36,724</td>
<td>34,852</td>
</tr>
<tr>
<td>(12,1.6,8)</td>
<td>63,642</td>
<td>63,741</td>
<td>57,750</td>
</tr>
</tbody>
</table>

Table 3: Computational results for the case where $n = 6$.

<table>
<thead>
<tr>
<th>prob.</th>
<th>up. bnd.</th>
<th>prob.</th>
<th>up. bnd.</th>
</tr>
</thead>
<tbody>
<tr>
<td>(6,1.2)</td>
<td>18,859</td>
<td>(12,1.2)</td>
<td>34,115</td>
</tr>
<tr>
<td>(6,1.4)</td>
<td>26,817</td>
<td>(12,1.4)</td>
<td>48,455</td>
</tr>
<tr>
<td>(6,1.8)</td>
<td>42,735</td>
<td>(12,1.8)</td>
<td>77,136</td>
</tr>
<tr>
<td>(6,1.2,2)</td>
<td>17,200</td>
<td>(12,1.2,2)</td>
<td>31,700</td>
</tr>
<tr>
<td>(6,1.2,4)</td>
<td>25,129</td>
<td>(12,1.2,4)</td>
<td>46,035</td>
</tr>
<tr>
<td>(6,1.2,8)</td>
<td>41,036</td>
<td>(12,1.2,8)</td>
<td>74,710</td>
</tr>
<tr>
<td>(6,1.6,2)</td>
<td>14,294</td>
<td>(12,1.6,2)</td>
<td>26,798</td>
</tr>
<tr>
<td>(6,1.6,4)</td>
<td>22,219</td>
<td>(12,1.6,4)</td>
<td>40,943</td>
</tr>
<tr>
<td>(6,1.6,8)</td>
<td>38,122</td>
<td>(12,1.6,8)</td>
<td>69,618</td>
</tr>
</tbody>
</table>

Table 4: Optimal objective value of problem (18)-(20).