Proceedings of the 2010 Winter Simulation Conference
B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, and E. Yücesan, eds.

SIMULATION-BASED METHODS FOR BOOKING CONTROL IN NETWORK REVENUE MANAGEMENT

Sumit Kunnumkal

Indian School of Business
Gachibowli, Hyderabad 500032, INDIA

Huseyin Topaloglu

School of Operations Research and Information Engineering
223 Rhodes Hall, Cornell University
Ithaca, NY 14853, USA

ABSTRACT

In this paper, we describe simulation-based stochastic approximation algorithms to find good bid price policies for booking control over an airline network. Our general approach visualizes the total expected profit as a function of the bid prices and searches for a good set of bid prices by using sample path derivatives of the total expected profit function. We demonstrate that the iterates of our stochastic approximation algorithms converge to a stationary point of the total expected profit function with probability one. Our computational experiments indicate that the bid prices computed by our approach perform quite well.

1 INTRODUCTION

Through the process of booking control, airlines decide which itinerary requests coming from their customers should be accepted and which itinerary requests should be rejected. What makes this process interesting is that the booking control decisions for an airline network interact with each other in a variety of ways. For example, an accepted itinerary request consumes capacities on a number of flight legs and there can be many other itineraries that consume capacities on these flight legs, indicating that there are interactions between the booking control decisions for different itineraries. Furthermore, the itinerary requests that we accept here and now have an influence on the itinerary requests that we can accept in the future, which implies that the booking control decisions at different time periods interact with each other as well. Thus, we need to have a view of the whole airline network and the whole selling horizon when making the booking control decisions. Furthermore, practical problem instances are often quite large, involving airline networks with tens of locations and selling horizons up to a year. Due these difficulties, it is often the case that airlines focus on a class of booking control policies parameterized by a number of parameters and look for a good booking control policy within this class. For example, protection level policies put upper bounds on how many seats should be reserved for different itineraries. These upper bounds are called protection levels. On the other hand, bid price policies associate an opportunity cost with a seat on each flight leg. These opportunity costs are called bid prices. An itinerary request is accepted only if the revenue from the itinerary request justifies the total opportunity cost of the seats that the itinerary request would consume. If we focus our attention to a protection level or a bid price policy, then the problem simply becomes that of searching for a good set of protection levels or bid prices.

In this paper, we develop stochastic approximation algorithms to search for a good set of bid prices. As mentioned above, a bid price policy associates a bid price with each flight leg in the airline network, capturing the opportunity cost of a seat. An itinerary request is accepted only if the revenue from the itinerary request exceeds the total opportunity cost of the seats that are consumed. Bid price policies have seen considerable attention from airlines as they are intuitive and easy to implement. There are a number of methods in the literature to compute good bid prices. Simpson (1989) and Williamson (1992) approximate the booking control problem by using a deterministic linear program that is formulated under the assumption that the demands for different itineraries take on their expected values. The authors use the shadow prices associated with the seat availability constraints in the linear program to compute bid prices. Talluri and van Ryzin (1998) provide some theoretical support for this linear program by showing that the bid prices obtained by the linear program are optimal in an asymptotic regime as the capacities on the flight legs and the length of the selling horizon increases linearly. Talluri and van Ryzin (1999) note that the linear program proposed by Simpson (1989) and Williamson (1992) ignores all uncertainty by assuming that the demands take on their expected values. They propose a randomized version of the linear program that works with sampled demand realizations rather

978-1-4244-9864-2/10/$26.00 ©2010 IEEE

than expected values. Bertsimas and Popescu (2003) extend the linear program to handle the case where overbooking is possible. Adelman (2007) formulates the booking control problem as a dynamic program with a high dimensional state variable. To deal with the problematic state variable, the author approximates the value functions with linear functions and finds good linear value function approximations. He shows that this approach can ultimately be used to compute bid prices. Similar to ours, there are a number of papers that find good booking control policies by using stochastic approximation algorithms. In particular, van Ryzin and Vulcano (2008a) and van Ryzin and Vulcano (2008b) focus on computing good protection limits by using stochastic approximation algorithms, whereas Topaloglu (2008) and Chaneton and Vulcano (2009) use stochastic approximation algorithms to compute good bid prices. Our work in this paper builds on Topaloglu (2008) and extends it in numerous directions. We give a unifying framework that can handle problems with overbooking, whereas Topaloglu (2008) exclusively considers the case where overbooking is not possible and all reservations show up at the departure time. Furthermore, Topaloglu (2008) focuses on convergence results for the proposed stochastic approximation algorithms, but we focus on an important implementation issue related to step size selection. We refer the reader to the book by Talluri and van Ryzin (2005) for detailed descriptions of other models and algorithms that can be used to find good bid prices for booking control.

The rest of the paper is organized as follows. In Section 2, we develop a stochastic approximation algorithm to find a good set of bid prices when overbooking is not possible. We begin by describing our notation and problem setup. We carefully explain how a bid price policy works. We develop a recursive expression that can be used to compute the total revenue obtained by a bid price policy for a single sample path of itinerary requests. We use this recursive expression to compute the sample path derivatives of the total revenue with respect to the bid prices. Finally, we use these sample path derivatives in a stochastic approximation algorithm to search for a good set of bid prices. In Section 3, we extend the development in Section 2 to the case where not all reservations show up at the departure time and overbooking is necessary. In Section 4, we provide computational experiments. In Section 5, we conclude.

2 NETWORK REVENUE MANGEMENT WITHOUT OVERBOOKING

We have a set of flight legs that can be used to serve the itinerary requests that arrive randomly over time. Whenever an itinerary request arrives into the system, we need to decide whether to accept or reject this itinerary request. An accepted itinerary request generates a revenue and consumes capacities on one or more flight legs. A rejected itinerary request simply leaves the system. Our objective is to find a policy to accept or reject the itinerary requests so as to maximize the total expected revenue.

2.1 Notation and Problem Formulation

The set of flight legs in the airline network is $\mathscr{L}$ and the set of itineraries is $\mathscr{J}$. If we accept a request for itinerary j, then we generate a revenue of $\tilde{f}_j$. To capture which itineraries use which flight legs, we let $\tilde{a}_{ij}$ take value one if itinerary j uses flight leg i, and take value zero otherwise. In this case, the vector $\tilde{a}_j = \{\tilde{a}_{ij} : i \in \mathscr{L}\}$ captures which flight legs are used by itinerary j. An itinerary is characterized by its revenue and the set of flight legs that it uses. In certain settings, an itinerary is referred to as an origin-destination-fare class combination. We prefer to use the itinerary terminology since the origin-destination-fare class terminology does not emphasize the fact that an itinerary is not only characterized by its origin and destination locations, but also by its through locations. We index the itinerary requests by $t = 1, \ldots, \tau$, where τ is a finite upper bound on the total number of itinerary requests that we may observe. We let x_{it} be the remaining capacity on flight leg i just before making the decision for itinerary request t and use $x_t = \{x_{it} : i \in \mathscr{L}\}$ to denote the vector of remaining capacities on the flight legs.

We capture the itinerary requests by the stochastic process $\tilde{\omega} = \{\tilde{J}_t : t = 1, \ldots, \tau\}$ with the interpretation that itinerary request t is for itinerary J_t. The stochastic process $\tilde{\omega}$ is quite natural to work with, but we use an alternative stochastic process to ease our notational burden. The only information that we need about an itinerary request is its revenue and the set of flight legs used by this itinerary request. In this case, if we let $f_t = \tilde{f}_{J_t}$ and $a_{it} = \tilde{a}_{i,J_t}$, then f_t and $a_t = \{a_{it} : i \in \mathscr{L}\}$ completely characterize all of the information that we need to know about itinerary request t. Therefore, we can use the stochastic process $\omega = \{(f_t, a_t) : t = 1, \ldots, \tau\}$ to characterize the arrival of the itinerary requests. For our purposes, the stochastic processes $\tilde{\omega}$ and ω are equivalent, but we work with the stochastic process ω throughout the paper. We note that a realization of the stochastic process ω completely characterizes $\{f_t : t = 1, \ldots, \tau\}$ and $\{a_t : t = 1, \ldots, \tau\}$, which are respectively the revenues that are associated with the itinerary requests and the $|\mathscr{L}|$-dimensional vectors that describe the flight legs used by the itinerary requests. To allow for the possibility that we do not have exactly τ itinerary requests in every sample path, we allow having $f_t = 0$ for all $t = \tau^0 + 1, \ldots, \tau$ for some random variable τ^0. In this case, it is not desirable to accept the last $\tau - \tau^0$ itinerary requests, which is equivalent to having τ^0 itinerary requests in a sample path.

2.2 Bid Price Policies

A bid price policy associates a bid price with each flight leg that characterizes the value of a seat on the flight leg. In this case, an itinerary request is accepted only if there is enough capacity on all of the relevant flight legs and the revenue from the itinerary request exceeds the sum of the bid prices associated with the flight legs that are in the requested itinerary. In particular, letting $\lambda = \{\lambda_i : i \in \mathscr{L}\}$ be the bid prices associated with the flight legs, the decision function for the policy characterized by bid prices λ can be written as

$$\tilde{u}_t(x_t, \lambda, \omega) = \mathbf{1}(x_t \geq a_t,\ f_t \geq \textstyle\sum_{i \in \mathscr{L}} a_{it}\, \lambda_i), \tag{1}$$

where $\mathbf{1}(\cdot,\cdot)$ is the indicator function. This is to say that $\tilde{u}_t(x_t, \lambda, \omega)$ takes value one if the policy characterized by bid prices λ accepts itinerary request t, and takes value zero otherwise. In the expression above, we take advantage of the fact that ω completely characterizes $\{f_t : t = 1, \ldots, \tau\}$ and $\{a_t : t = 1, \ldots, \tau\}$ so that ω in the argument of the decision function on the left provides a_t in the argument of the indicator function on the right. The first argument of the indicator function ensures that there is enough capacity on all of the flight legs, whereas the second argument ensures that the revenue from the itinerary request exceeds the sum of the bid prices associated with the flight legs that are in the requested itinerary.

We can compute the total revenue obtained from itinerary requests $t, \ldots, \tau$ by using the revenue function recursively defined by

$$\tilde{R}_t(x_t, \lambda, \omega) = f_t\, \tilde{u}_t(x_t, \lambda, \omega) + \tilde{R}_{t+1}(x_t - a_t\, \tilde{u}_t(x_t, \lambda, \omega), \lambda, \omega), \tag{2}$$

where the first expression on the right side corresponds to the revenue from itinerary request t, whereas the second expression corresponds to the total revenue from itinerary requests $t+1, \ldots, \tau$. If the initial capacities on the flight legs are given by x_1, then the random variable $\tilde{R}_1(x_1, \lambda, .\omega)$ captures the total revenue obtained from all of the itinerary requests. Since our objective is to find a set of bid prices to maximize the total expected revenue from all of the itinerary requests, we can solve the problem

$$\max_{\lambda}\ \mathbb{E}\{\tilde{R}_1(x_1, \lambda, \omega)\}. \tag{3}$$

Unfortunately, due to the indicator function in (1), the decision function is not differentiable with respect to the bid prices. This implies that the revenue function in (2) is not differentiable with respect to the bid prices either. This observation precludes us from solving problem (3) by using sample path derivatives of the revenue function. In the next section, we resolve this difficulty by formulating an appropriately smoothed version of problem (3).

2.3 Smoothed Revenue Function

In this section, we formulate an appropriately smoothed version of problem (3) that we can solve by using sample path derivatives of its objective function. We consider a bid price 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 that are in the requested itinerary. To this end, we let $\theta(\cdot)$ be a differentiable and increasing function satisfying $\lim_{p \to -\infty} \theta(p) = 0$ and $\lim_{p \to \infty} \theta(p) = 1$. We assume that $\theta(\cdot)$ and its derivative $\dot{\theta}(\cdot)$ are Lipschitz continuous. In the smoothed version of problem (3), we can try using the decision function

$$\hat{u}_t(x_t, \lambda, \omega) = \min\Big\{ \min\{x_{it} : i \in \mathscr{L}_t^+\}, \theta(f_t - \textstyle\sum_{i \in \mathscr{L}} a_{it}\, \lambda_i) \Big\},$$

where we let $\mathscr{L}_t^+ = \{i \in \mathscr{L} : a_{it} = 1\}$. The decision function above can still be problematic as it is nondifferentiable when the arguments of the min operators are equal to each other. To overcome difficulty, we perturb the remaining capacities by small random amounts. In particular, letting $\alpha = \{\alpha_{it} : i \in \mathscr{L},\ t = 1, \ldots, \tau\}$ be random variables that are uniformly distributed over the small interval $[0, \varepsilon]$, we modify the last decision function as

$$u_t(x_t, \lambda, \omega) = \min\Big\{ \min\{x_{it} + \alpha_{it} : i \in \mathscr{L}_t^+\}, \theta(f_t - \textstyle\sum_{i \in \mathscr{L}} a_{it}\, \lambda_i) \Big\}, \tag{4}$$

where we redefine the stochastic process as $\omega = \{(f_t, a_t, \alpha_t) : t = 1, \ldots, \tau\}$ with $\alpha_t = \{\alpha_{it} : i \in \mathscr{L}\}$ to include the random capacity perturbations. Assuming that the capacity perturbations $\{\alpha_{it} : i \in \mathscr{L},\ t = 1, \ldots, \tau\}$ are independent, the arguments of the min operators in (4) are distinct with probability one (w.p.1), which implies that the decision function in (4) is differentiable w.p.1. Using the decision function in (4), the revenue function becomes

$$R_t(x_t,\lambda,\omega) = f_t\, u_t(x_t,\lambda,\omega) + R_{t+1}(x_t + \alpha_t - a_t\, u_t(x_t,\lambda,\omega),\lambda,\omega). \tag{5}$$

Therefore, we can find a good set of bid prices by solving the problem

$$\max_{\lambda} \quad \mathbb{E}\{R_1(x_1,\lambda,\omega)\}. \tag{6}$$

Noting the fact that the decision function in (4) is differentiable w.p.1 and using backward induction over the itinerary requests, one can show that $R_1(x_1,\lambda,\omega)$ is differentiable with respect to the bid prices w.p.1. This observation ultimately allows us to solve problem (6) by using sample path derivatives of the revenue function. In the next section, we show how to compute sample path derivatives of the revenue function.

2.4 Sample Path Derivatives of Smoothed Revenue Function

We begin this section by defining some notation. We use $\partial_i^\Lambda R_t(x_t,\lambda,\omega)$ to denote the derivative of $R_t(\cdot,\cdot,\omega)$ with respect to the bid price associated with flight leg i evaluated at bid prices λ and remaining leg capacities x_t. Similarly, we use $\partial_i^X R_t(x_t,\lambda,\omega)$ to denote the derivative of $R_t(\cdot,\cdot,\omega)$ with respect to the remaining capacity on flight leg i evaluated at bid prices λ and remaining leg capacities x_t. That is, we have

$$\partial_i^\Lambda R_t(x_t,\lambda,\omega) = \left.\frac{\partial R_t(z_t,\gamma,\omega)}{\partial \gamma_i}\right|_{(z_t,\gamma)=(x_t,\lambda)} \quad \text{and} \quad \partial_i^X R_t(x_t,\lambda,\omega) = \left.\frac{\partial R_t(z_t,\gamma,\omega)}{\partial z_{it}}\right|_{(z_t,\gamma)=(x_t,\lambda)}.$$

We use the notation $\partial_i^\Lambda u_t(x_t,\lambda,\omega)$ and $\partial_i^X u_t(x_t,\lambda,\omega)$ with similar interpretation.

To compute the sample path derivative of the objective function of problem (6) with respect to the bid price of flight leg i, we differentiate (5) with respect to the bid price of flight leg i and use the chain rule to obtain

$$\begin{aligned}\partial_i^\Lambda R_t(x_t,\lambda,\omega) = {} & f_t\, \partial_i^\Lambda u_t(x_t,\lambda,\omega) + \partial_i^\Lambda R_{t+1}(x_t + \alpha_t - a_t\, u_t(x_t,\lambda,\omega),\lambda,\omega) \\ & - \sum_{k\in\mathscr{L}_t^+} \partial_k^X R_{t+1}(x_t + \alpha_t - a_t\, u_t(x_t,\lambda,\omega),\lambda,\omega)\, \partial_i^\Lambda u_t(x_t,\lambda,\omega),\end{aligned} \tag{7}$$

where we use the fact that $a_{it} = 1$ if $i \in \mathscr{L}_t^+$ and $a_{it} = 0$ if $i \notin \mathscr{L}_t^+$. To compute the terms on the right side above, differentiating (5) with respect to the remaining capacity on flight leg i and using the chain rule yield

$$\begin{aligned}\partial_i^X R_t(x_t,\lambda,\omega) = {} & f_t\, \partial_i^X u_t(x_t,\lambda,\omega) + \partial_i^X R_{t+1}(x_t + \alpha_t - a_t\, u_t(x_t,\lambda,\omega),\lambda,\omega) \\ & - \sum_{k\in\mathscr{L}_t^+} \partial_k^X R_{t+1}(x_t + \alpha_t - a_t\, u_t(x_t,\lambda,\omega),\lambda,\omega)\, \partial_i^X u_t(x_t,\lambda,\omega).\end{aligned} \tag{8}$$

Differentiating (4) with respect to the bid price of flight leg i, we have

$$\partial_i^\Lambda u_t(x_t,\lambda,\omega) = \begin{cases} -a_{it}\, \dot{\theta}(f_t - \sum_{i\in\mathscr{L}} a_{it}\,\lambda_i) & \text{if } \theta(f_t - \sum_{i\in\mathscr{L}} a_{it}\,\lambda_i) \le \min\{x_{it} + \alpha_{it} : i \in \mathscr{L}_t^+\} \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

We emphasize that the expression above is correct only in w.p.1 sense. In particular, the decision function in (4) is not differentiable when the arguments of the min operators are equal to each other, but this event occurs with probability zero. Finally, we differentiate (4) with respect to the remaining capacity on flight leg i to obtain

$$\partial_i^X u_t(x_t,\lambda,\omega) = \begin{cases} 1 & \text{if } i \in \mathscr{L}_t^+ \text{ and } x_{it} + \alpha_{it} \le \min\Big\{\min\{x_{kt} + \alpha_{kt} : k \in \mathscr{L}_t^+ \setminus \{i\}\},\, \theta(f_t - \sum_{i\in\mathscr{L}} a_{it}\,\lambda_i)\Big\} \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

Similar to (9), the expression above is correct only in w.p.1 sense.

We can use (7)-(10) to compute the sample path derivative of the objective function of problem (6). In particular, we can simulate the evolution of the system with bid prices λ under sample path ω. In this case, the remaining capacities on the flight legs after each itinerary request are recursively given by $X_{t+1}(\lambda,\omega) = X_t(\lambda,\omega) + \alpha_t - a_t\, u_t(X_t(\lambda,\omega),\lambda,\omega)$ with the initial condition that $X_1(\lambda,\omega) = x_1$. We can use (9) and (10) to respectively compute $\partial_i^\Lambda u_t(X_t(\lambda,\omega),\lambda,\omega)$ and $\partial_i^X u_t(X_t(\lambda,\omega),\lambda,\omega)$ for all $t = 1,\ldots,\tau$. Following this computation, we can use (8) and move backwards over the itinerary requests to compute $\partial_i^X R_t(X_t(\lambda,\omega),\lambda,\omega)$ for all $t = 1,\ldots,\tau$. Finally, we can use (7) and move backwards over the itinerary requests to compute $\partial_i^\Lambda R_t(X_t(\lambda,\omega),\lambda,\omega)$ for all $t = 1,\ldots,\tau$. In this case, $\partial_i^\Lambda R_1(X_1(\lambda,\omega),\lambda,\omega) = \partial_i^\Lambda R_1(x_1,\lambda,\omega)$ gives the sample path derivative of the objective function of problem (6) with respect to the bid price of flight leg i. In the next section, we give an algorithm that uses this sample path derivative to solve problem (6).

2.5 Stochastic Approximation Algorithm

The following algorithm can be used to solve problem (6) by using sample path derivatives of the objective function of this problem.

1. Choose the initial bid prices $\lambda^1 = \{\lambda_i^1 : i \in \mathscr{L}\}$ arbitrarily. Initialize the iteration counter n to 1.
2. Generate a sample path ω^n at iteration n. Compute $\partial_i^\Lambda R_1(x_1, \lambda^n, \omega^n)$ for all $i \in \mathscr{L}$ by using (7)-(10).
3. Letting σ^n be the step size at iteration n, update the bid prices as $\lambda_i^{n+1} = \lambda_i^n + \sigma^n \, \partial_i^\Lambda R_1(x_1, \lambda^n, \omega^n)$ for all $i \in \mathscr{L}$.
4. Increase n by one and go to Step 2.

The algorithm above is a standard stochastic approximation algorithm driven by the sample path derivatives $\{\partial_i^\Lambda R_1(x_1, \lambda^n, \omega^n) : i \in \mathscr{L}\}$ at iteration n. We let $\mathscr{F}^n = (\lambda^1, \omega^1, \ldots, \omega^{n-1})$ be the history of the algorithm up to iteration n. We assume that the distribution of ω^n conditional on $\mathscr{F}^n$ is the same as the distribution of ω. We have the following convergence result for the algorithm above.

Proposition 1. *Assume that the sequence of bid prices $\{\lambda^n\}_n$ are generated by the algorithm above and the sequence of step sizes satisfy $\sigma^n \geq 0$ for all $n = 1, 2, \ldots$, $\sum_{n=1}^\infty \sigma^n = \infty$ and $\sum_{n=1}^\infty [\sigma^n]^2 < \infty$. In this case, every limit point λ^* of the sequence $\{\lambda^n\}_n$ is a stationary point of the objective function of problem* (6) *satisfying $\mathbb{E}\{\partial_i^\Lambda R_1(x_1, \lambda^*, \omega)\} = 0$ for all $i \in \mathscr{L}$.*

The convergence result in Proposition 1 provides theoretical support for the algorithm above. We emphasize that the objective function of problem (6) is not necessarily concave and the stationary point mentioned in Proposition 1 can be a local maximum, a local minimum or an inflection point. Nevertheless, our computational results indicate that the bid prices obtained by the algorithm above perform remarkably well.

3 NETWORK REVENUE MANGEMENT WITH OVERBOOKING

The ideas in Section 2 can be extended to the case where overbooking is possible. Similar to Section 2, we consider a setting where the itinerary requests arrive randomly over time and we need to decide whether to accept or reject each itinerary request. However, only a portion of the accepted itinerary requests show up at the departure time. If we cannot accommodate all of the accepted itinerary requests that show up at the departure time, then we incur a penalty cost for denied boarding. Our objective is to find a policy to accept or reject the itinerary requests on the flight legs so as to maximize the total expected profit. In this setting, the total expected profit is the difference between the total expected revenue from the accepted itinerary requests and the total expected cost of denied boarding.

We describe the main points in which our notation deviates from the one in Section 2.1. We work with the stochastic process $\omega = \{J_t : t = 1, \ldots, \tau\}$ with the interpretation that itinerary request t is for itinerary J_t. The policy characterized by bid prices λ accepts a request for itinerary j 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. Since only a portion of the accepted itinerary requests show up at the departure time, if overbooking is allowed, then the accepted itinerary requests can exceed the capacities on the flight legs . Thus, we do not need to adhere to the capacity limitations when accepting or rejecting the itinerary requests. In this case, the decision function for the policy characterized by bid prices λ becomes $\tilde{u}_t(\lambda, \omega) = \mathbf{1}(\tilde{f}_{J_t} \geq \sum_{i \in \mathscr{L}} \tilde{a}_{i,J_t} \lambda_i)$. Similar to the decision function in (1), the indicator function in the last expression is not convenient to work with as it introduces a source of nondifferentiability. Therefore, we assume that 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 that are in the requested itinerary. This reasoning yields the decision function

$$u_t(\lambda, \omega) = \theta(\tilde{f}_{J_t} - \textstyle\sum_{i \in \mathscr{L}} \tilde{a}_{i,J_t} \lambda_i),$$

where $\theta(\cdot)$ satisfies the same conditions stated in Section 2.3. To capture the state of the system, we let z_{jt} be the total number of accepted requests for itinerary j just before making the decision for itinerary request t and use $z_t = \{z_{jt} : j \in \mathscr{J}\}$ to denote the vector of accepted itinerary requests. In this case we can accumulate the total profit obtained from the itinerary requests $t, \ldots, \tau$ by using the revenue function recursively defined by

$$R_t(z_t, \lambda, \omega) = \tilde{f}_{J_t} \, u_t(\lambda, \omega) + R_{t+1}(z_t + e_{J_t} \, u_t(\lambda, \omega), \lambda, \omega),$$

where e_j is the $|\mathscr{J}|$-dimensional unit vector with a one in the element corresponding to $j \in \mathscr{J}$. The boundary condition of the recursive expression above is $R_{\tau+1}(z_{\tau+1}, \lambda, \omega) = -F(z_{\tau+1})$, where $F(z_{\tau+1})$ is the total expected penalty cost incurred at the departure time given that we arrive at the departure time with $z_{\tau+1}$ accepted itinerary requests. We leave the form of $F(\cdot)$ unspecified since different airlines decide how to deny boarding in different ways, but it is worthwhile to emphasize that $F(\cdot)$ depends on the capacities on the flight legs, the compensation amounts paid to denied boardings and the probability that an accepted itinerary request shows up at the departure time. We only assume that $F(z_{\tau+1})$ and each component of its gradient are Lipschitz continuous. Depending on the exact expression for $F(\cdot)$, satisfying the Lipschitz continuity assumption may require adding random perturbations to $z_{\tau+1}$ similar to the ones in Section 2.3. Kunnumkal and Topaloglu (2009) give some reasonable expressions for $F(\cdot)$. By using an approach that closely follows the one in Section 2.4, it is possible to derive recursive expressions to compute sample path derivatives of $R_1(x_1,\lambda,\omega)$. This allows us to construct a stochastic approximation algorithm similar to the one in Section 2.5 that finds a good set of bid prices. One can establish a convergence result similar to the one in Proposition 1 for this stochastic approximation algorithm.

4 COMPUTATIONAL EXPERIMENTS

In this section, we provide computational experiments that test the quality of the bid prices obtained by the algorithm in Section 2.5.

4.1 Computational Setup

In our computational experiments, we focus on an airline network composed of one hub location serving six spoke locations. There is a flight from the hub location to each spoke location and a flight from each spoke location to the hub location. Therefore, there are 12 flight legs in the airline network. There is a high fare and a low fare itinerary associated with each origin destination pair in the airline network. Since there are seven locations, we have $7\times 6\times 2 = 84$ itineraries. We denote the fare difference between high fare and low fare itineraries by κ. In other words, high fare itineraries are κ times more expensive than low fare itineraries. We use γ to denote the ratio between the total expected number of seats consumed by all itinerary requests and the total number of seats available on the flight legs. Therefore, the parameter γ is an indicator of how tightly constrained the capacities on the flight legs are when compared with the total expected demand.

We use the deterministic linear program that we mention in Section 1 as a benchmark strategy in our computational experiments. This benchmark strategy assumes that the requests for all itineraries take on their expected values and formulates the booking control problem as a deterministic linear program. Noting the definition of J_t in Section 2.1, the total expected demand for itinerary j is given by $\sum_{t=1}^{\tau}\mathbb{E}\{\mathbf{1}(j=J_t)\}$. Using the decision variable w_j to denote the number of requests for itinerary j that we plan to accept, we can solve the linear program

$$\begin{aligned} \max \quad & \sum_{j\in\mathscr{J}} \tilde{f}_j\, w_j \\ \text{subject to} \quad & \sum_{j\in\mathscr{J}} \tilde{a}_{ij}\, w_j \le x_{i1} \qquad i\in\mathscr{L} \\ & 0 \le w_j \le \sum_{t=1}^{\tau}\mathbb{E}\{\mathbf{1}(j=J_t)\} \qquad j\in\mathscr{J} \end{aligned}$$

to estimate the total expected revenue. The first set of constraints ensure that the itinerary requests that we plan to accept do not violate the leg capacities, whereas the second set of constraints ensure that the itinerary requests that we plan to accept do not exceed the expected numbers of itinerary requests. In this case, letting $\{\mu_i : i\in\mathscr{L}\}$ be the optimal values of the dual variables associated with the first set of constraints, μ_i captures the opportunity cost of a seat on flight leg i. Therefore, we can use μ_i as the bid price associated with flight leg i. Throughout this section, we refer to this benchmark strategy as DLP, standing for deterministic linear program. We refer to the stochastic approximation algorithm in Section 2.5 as SA. Unless otherwise stated, we terminate this algorithm after 10,000 iterations and use the step size $\sigma^n = 20/(40+n)$ at iteration n.

4.2 Computational Results

In Table 1, we compare the performances of SA and DLP. The columns of this table correspond to different test problems indexed by (κ,γ), where κ and γ are as described in Section 4.1. The second and third rows respectively show the total expected revenues obtained by SA and DLP. We estimate these total expected revenues by simulating the decision trajectories for SA and DLP for multiple sample paths. We use common random numbers when simulating the decision trajectories for the two benchmark strategies. The fourth row shows the percent gap between the total expected revenues obtained by SA and DLP. Overall, the results indicate that SA can improve on the performance of DLP with significant margins.

Table 1: Performances of SA and DLP.

Benchmark	Test problem (κ,γ)			
Strategy	$(1.2,4)$	$(1.2,8)$	$(1.6,4)$	$(1.6,8)$
SA	22,773	36,758	19,746	33,579
DLP	21,208	33,269	17,771	28,628
Perc. Gap	6.87%	9.49%	10.00%	14.74%

One of the concerns in using stochastic approximation algorithms is that their performance can be dependent on the choice of the step size. Our experience has been that while some tuning of the step size is necessary, once we get in a close neighborhood of a reasonable step size rule, fluctuations around this step size rule do not create a big problem. In all of our computational experiments, we use a step size of the form $\sigma^n = a/(b+n)$ at iteration n. We obtain the computational results in Table 1 by using $a = 20$ and $b = 40$. In Table 2, we show the performance of SA with different choices for (a,b). The columns of this table correspond to different test problems indexed in the same fashion as in Table 1. Each row shows the total expected revenues obtained by SA by using a particular choice of (a,b). Noting that $20/(40+1) \approx 200/(400+1) \approx 2000/(4000+1) \approx 0.5$, the step sizes at the first iteration when $(a,b) \in \{(20,40),(200,400),(2000,4000)\}$ are roughly the same. However, we have $100\times[20/(40+10000)] \approx 10\times[200/(400+10000)] \approx 2000/(4000+10000)$ and the step sizes at the final iteration differ roughly by a factor of ten. Similar observations apply to the case where $(a,b) \in \{(80,40),(800,400),(8000,4000)\}$. Overall, the results in Table 2 indicate that the performance of SA remains relatively stable for different values of (a,b). Furthermore, a step size rule that appears to perform reasonably well for one test problem continues to perform well for other test problems as well.

Table 2: Performance of SA with different step sizes.

	Test problem (κ,γ)			
(a,b)	$(1.2,4)$	$(1.2,8)$	$(1.6,4)$	$(1.6,8)$
$(20,40)$	22,773	36,758	19,746	33,579
$(200,400)$	22,715	37,043	19,680	33,874
$(2000,4000)$	22,576	37,115	19,663	34,404
$(80,40)$	22,792	36,571	19,753	33,225
$(800,400)$	22,719	37,141	19,728	33,710
$(8000,4000)$	22,689	36,712	19,726	33,711

5 CONCLUSIONS

In this paper, we developed a stochastic approximation algorithm to make booking control decisions over an airline network. Our computational results indicated that our stochastic approximation algorithm can provide significant improvements over a benchmark strategy that ignores the randomness in demand altogether. The demand model that we use in this paper assumes that each customer arrives into the system with the intention of purchasing a fixed itinerary. The only decision is to whether accept or reject this itinerary request. An important avenue for further investigation is to see whether it is possible to develop stochastic approximation algorithms for the case where the customers do not arrive into the system with the intention of purchasing a fixed itinerary, but they observe the set of itineraries available for purchase and make a choice among them. Models and algorithms that capture the customer choice process are crucial to improve the practical applicability of network revenue management tools.

REFERENCES

Adelman, D. 2007. Dynamic bid-prices in revenue management. *Operations Research* 55 (4): 647–661.

Bertsimas, D., and I. Popescu. 2003. Revenue management in a dynamic network environment. *Transportation Science* 37:257–277.

Chaneton, J., and G. Vulcano. 2009. Computing bid-prices for revenue management under customer choice behavior. Technical report, New York University, Stern School of Business.

Kunnumkal, S., and H. Topaloglu. 2009. A stochastic approximation algorithm to compute bid prices for joint capacity allocation and overbooking over an airline network. Technical report, Cornell University, School of Operations Research and Information Engineering.

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

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

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

Talluri, K. T., and G. J. van Ryzin. 2005. *The theory and practice of revenue management*. New York, NY: Springer.

Topaloglu, H. 2008. A stochastic approximation method to compute bid prices in network revenue management problems. *INFORMS Journal on Computing* 20 (4): 596–610.

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

van Ryzin, G., and G. Vulcano. 2008b. Simulation-based optimization of virtual nesting controls for network revenue management. *Operations Research* 56 (4): 865–880.

Williamson, E. L. 1992. *Airline network seat control*. Ph. D. thesis, Massachusetts Institute of Technology, Cambridge, MA.

AUTHOR BIOGRAPHIES

SUMIT KUNNUMKAL is an Assistant Professor at the Indian School of Business. He holds a Ph.D. in Operations Research from Cornell University. His current research areas are network revenue management and supply chain management. His research combines techniques from dynamic programming, stochastic optimization and simulation to develop algorithms to improve decision making in revenue management and supply chain systems. His webpage is at <http://www.isb.edu/faculty/facultydir.aspx?ddlFaculty=211>.

HUSEYIN TOPALOGLU is an Associate Professor in the School of Operations Research and Information Engineering at Cornell University. He holds a B.Sc. in Industrial Engineering from Bogazici University in Turkey, and a Ph.D. in Operations Research and Financial Engineering from Princeton University. His research interests include stochastic programming and approximate dynamic programming with applications in transportation logistics, revenue management and supply chain management. He teaches courses on dynamic programming, simulation modeling, systems engineering and revenue management. His webpage is at <http://people.orie.cornell.edu/~huseyin>.