Re-Solving Stochastic Programming Models for Airline Revenue Management *

Lijian Chen
Department of Industrial, Welding and Systems Engineering
The Ohio State University
Columbus, OH 43210
chen.855@osu.edu

Tito Homem-de-Mello
Department of Industrial Engineering and Management Sciences
Northwestern University
Evanston, IL 60208
tito@northwestern.edu

June 22, 2006

Abstract

We study some mathematical programming formulations for the origin-destination model in airline revenue management. In particular, we focus on the traditional probabilistic model proposed in the literature. The approach we study consists of solving a sequence of two-stage stochastic programs with simple recourse, which can be viewed as an approximation to a multi-stage stochastic programming formulation to the seat allocation problem. Our theoretical results show that the proposed approximation is robust, in the sense that solving more successive two-stage programs can never worsen the expected revenue obtained with the corresponding allocation policy. Although intuitive, such a property is known not to hold for the traditional deterministic linear programming model found in the literature. We also show that this property does not hold for some bid-price policies. In addition, we propose a heuristic method to choose the re-solving points, rather than re-solving at equally-spaced times as customary. Numerical results are presented to illustrate the effectiveness of the proposed approach.

*Supported by the National Science Foundation under grant DMI-0115385.

1 Introduction

Revenue management involves the application of quantitative techniques to improve profits by controlling the prices and availabilities of various products that are produced with scarce resources. Perhaps the best known revenue management application occurs in the airline industry, where the products are tickets (for itineraries) and the resources are seats on flights. In view of many successful applications of revenue management in different areas, this topic has received considerable attention in the past few years both from practitioners and academics. The recent book by Talluri and Van Ryzin [20] provides a comprehensive introduction to this field, see also references therein.

A common way to model the airline booking process is as a sequential decision problem over a fixed time period, in which one decides whether each request for a ticket should be accepted or rejected. A typical assumption is that one can separate demand for individual *itinerary-class pairs*; that is, each request is for a particular class on a particular itinerary, and yields a pre-specified fare. Typically, a class is determined by particular constraints associated with the ticket rather than the physical seat. For example, a certain class may require a 14-day advance purchase, or a Saturday night stay, etc. (for the purposes of this paper, we assume that first-class seats are allocated separately).

The existence of different classes reflects different customer behaviors. The classical example is that of customers traveling for leisure and those traveling on business. The former group typically books in advance and is more price-sensitive, whereas the latter behaves in the opposite way. Airline companies attempt to sell as many seats as possible to high-fare paying customers and at the same time avoid the potential loss resulting from unsold seats. In most cases, rejecting an early (and lower-fare) request saves the seat for a later (and higher-fare) booking, but at the same time that creates the risk of flying with empty seats. On the other hand, accepting early requests raises the percentage of occupation but creates the risk of rejecting a future high-fare request because of the constraints on capacity.

Many of early models were built for single flights. While that environment allows for the derivation of optimal policies via dynamic programming even with the incorporation of extra features [19], the drawback is clear in that the booking policy is only locally optimized and it cannot guarantee global optimality. Because of that, models that can incorporate a *network* of flights are usually preferred by the airlines. Network models, however, can only provide heuristics for the booking process, since determining the optimal action for each request in a network environment is impractical from a computational point of view. One type of heuristics is based on mathematical programs, where the decision variables are the number of seats to allocate to each class. In particular, methods based on *linear programming* techniques have been very popular in industry, for several reasons: first, linear programming is a well developed method in operations research; its properties have been thoroughly studied for decades. Secondly, commercial software packages for linear programming are widely available and have proven efficient and reliable in practice. Finally, the dual information obtained from the linear program can be used to derive alternative booking policies, based on *bid prices*; we will return to that in section 4.

Booking methods based on linear programming were thoroughly investigated by Williamson [25]. The basic models take stochastic demand into account only through expected values, thus yielding a deterministic program that can be easily solved. However, the drawback of such approach is obvious, as it ignores any distributional information about the demand. A common way to attempt to overcome that problem is to *re-solve* the LP several times during the booking horizon. While such an approach may seem intuitive, it turns out that re-solving can actually backfire — indeed, Cooper [7] shows a counter-example where re-solving the LP model may lower the total expected revenue.

An alternative way to incorporate demand distribution information into the model is by formulating a *stochastic* linear program. In the particular case of airline bookings, such models typically reduce to simple recourse models, a formulation that is called *probabilistic nonlinear program* in the revenue management literature (see, e.g., [25]). Higle and Sen [13] propose a different stochastic programming model, based on leg-based seat allocations, which yields an alternative way to compute bid prices. Another stochastic optimization model is proposed by Cooper and Homem-de-Mello [8], who consider a hybrid method where the second stage is actually the optimal value function of a dynamic program.

In this paper we discuss various aspects of the multi-stage version of the simple recourse model discussed above (henceforth denoted respectively MSSP and SLP). The MSSP model we present is shown to yield a better policy than SLP in terms of expected total revenue under the corresponding allocation policies; however, that multi-stage model does not have convexity properties (even its continuous relaxation), whereas simple recourse models can be solved very efficiently with linear integer programming. Of course, these conclusions are valid for the underlying MSSP model; alternative multi-stage models proposed in the literature (notably the ones in DeMiguel and Mishra [10] and Möller et al. [16]) do not suffer from the non-concavity issue, although a precise relationship with the SLP model is not established in those papers.

Given the difficulty to develop exact algorithms for large multi-stage problems, we propose an approximation based on solving a *sequence* of two-stage simple recourse models. The main advantage of such an approach is that, as mentioned above, each two-stage problem can be solved very efficiently, so an approximating solution to the MSSP can be obtained reasonably quickly. The idea of solving two-stage problems sequentially is not new, and appears in the literature under names such as *rolling horizon* and *rolling forward*; see, for instance, [1, 2, 15]. The details on the implementation of the rolling horizon, however, vary in the above works. Our work is more closely related to Balasubramanian and Grossmann [1] in that we consider *shrinking horizons*, i.e., each two-stage problem is solved over a period spanning from the current time until the end of the booking horizon. In this paper this is called the *re-solving SLP* approach.

Although the rolling horizon approach has been proposed in the literature, to the best of our knowledge there have been no analytical results regarding the quality of the approximation. In this paper we provide some results of that nature, though we do not claim to give definitive answers. More specifically, we compare the policies obtained from the re-solving SLP approach with the policies from the MSSP model. We show that, for a given partition into stages, the policy from MSSP is better than the policy from re-solving SLP. However, the inclusion of just one extra re-solving point can make the re-solving approach better. The importance of this conclusion arises from the fact that, because of the sequential nature of the re-solving procedure, adding an extra re-solving point requires little extra computational effort; in comparison, including an extra stage in a multi-stage model makes the problem considerably bigger and therefore harder to solve.

We also study the effect of re-solving the SLP model, compared to not re-solving it. Our results show that, unlike the aforementioned example in [7] for the DLP model, solving the SLP sequentially cannot be worse (in terms of expected revenue from the resulting policy) than solving it only once. In addition, we provide an example to illustrate that re-solving may actually be worse in the context of standard bid-price policies, where the bid prices are calculated from the dual problem of either the DLP or the SLP models. These results are, to the best of our knowledge, novel.

Motivated by the flexibility of the re-solving approach, we also study the issue of whether one can improve the results by carefully choosing the re-solving points instead of using equally-sized intervals as it is usually done. Indeed, the structure of our problem allows us to do so, and we provide a heuristic algorithm to determine the re-solving points. Our numerical results, run for two relatively small-sized networks, indicate that the procedure is effective.

The remainder of the paper is organized as follows: in section 2 we introduce the notation and describe mathematical programming methods for the seat allocation problem. The re-solving approach is treated in detail in section 3, and bid-price policies are discussed in section 4. Section 5 describes the algorithm for improving the choice of re-solving points, whereas section 6 presents numerical results. Concluding remarks are presented in section 7.

2 Allocation methods

Following the standard models in the literature, we consider a network of flights involving p booking classes of customers. This model can represent demand for a network of flights that depart within a particular day. Each customer requests one out of n possible itineraries, so we have $r := np$ itinerary-fare class combinations. The booking process is realized over a time horizon of length τ. Let $\{N_{jk}(t)\}$ denote the point process generated by the arrivals of class-k customers who request itinerary j. Typical cases customary in the revenue management literature are (i) $\{N_{jk}(t)\}$ is a (possibly nonhomogeneous) Poisson process, and (ii) there is at most one unit of demand per time period. Throughout this paper we do not make those assumptions unless when explicitly said otherwise; we only assume that arrival processes corresponding to different pairs (j, k) are independent of each other.

The demand for itinerary-class (j, k) over the whole horizon is denoted by ξ_{jk} (i.e., $\xi_{jk} = N_{jk}(\tau)$) and we denote by ξ the whole vector (ξ_{jk}). The network is comprised of m leg-cabin combinations, with capacities $c := (c_1, \ldots, c_m)$, and is represented by an $(m \times np)$-matrix $A \equiv (a_{i,jk})$. The entry $a_{i,jk} \in \{0, 1\}$ indicates whether class k customers use leg i in itinerary j.

Most policies we deal with in this paper are of allocation type. We denote by x_{jk} the decision variable corresponding to the number of seats to be allocated to class k in itinerary j. Whenever a itinerary-class pair (j, k) is accepted, the revenue corresponding to the fare f_{jk} accrues. A customer's request is rejected if no seats are available for his itinerary-class, in which case no revenue is realized. The vectors of decision variables and fares are denoted respectively by $x = (x_{jk})$ and $f = (f_{jk})$.

Allocation methods require solving an optimization problem to find initial allocations before the booking process starts. The classical deterministic linear program (DLP) for the network problem is written as follows:

$$\max\,\{f^T x \,:\, Ax \leq c,\ x \leq \mathbb{E}[\xi],\ x \geq 0\}. \tag{DLP}$$

Model [DLP] is well known in the revenue management literature. Implementation of the resulting policy x^* is very simple — accept at most x^*_{jk} class k customers in itinerary j. Notice that the policy is well-defined even if the solution x^* is not integer. Notice also that the objective function of [DLP] is *not* the actual revenue resulting from the policy x^*, even if x^* is integer.

A major drawback of formulation [DLP] is the fact that it completely ignores any information regarding the distribution of demand, except for the mean. This leads to the stochastic programming formulation

$$\max\,\{f^T \mathbb{E}[\min\{x, \xi\}] \,:\, Ax \leq c,\ x \in \mathbb{Z}_+\}.$$

where the *min* operation is interpreted componentwise (note that we have imposed an integrality constraint on x to ensure we get an integer allocation of seats; we will comment on that later). Equivalently, we have

$$\begin{aligned} & \max\ f^T x + \mathbb{E}[Q(x, \xi)] \\ \text{subject to:} & \\ & Ax \leq c, \\ & x \in \mathbb{Z}_+, \end{aligned} \tag{SLP}$$

where $Q(x,\xi) = \max \{-f^T y | x - y \le \xi,\ y \ge 0\}$. Notice that [SLP] can be formulated as a two-stage integer problem with *simple recourse*. A major advantage of such models is that, when ξ has a discrete distribution with finitely many scenarios, problem [SLP] can be easily solved because of its special structure. In principle, this may not be the case of our model, for example when the total demand for each itinerary-class pair (j,k) has Poisson distribution, which has infinite support. It is clear, however, that in that case all but a finite number of points in the distribution have negligible probability; thus, we can approximate the distribution of ξ_{jk} by a *truncated* Poisson. Thus, in what follows we assume that ξ takes on finitely many values, and that those values are integer.

To describe the solution procedure, we need to introduce some notation. For each itinerary-class pair (j,k), let S_{jk} denote the number of possible values taken by ξ_{jk}, and let $d^1_{jk}, \ldots, d^{S_{jk}}_{jk}$ denote those values, ordered from lowest to highest. Let $\{\delta^s_{jk}\}$, $s = 1, \ldots, S_{jk}$ be coefficients defined as $\delta^s_{jk} := f_{jk} P(\xi_{jk} \le d^s_{jk})$. As discussed in [5] and [14], problem [SLP] can then be re-written as

$$
\begin{aligned}
\max_{x,u,u^0}\ & f^T x - \sum_{j,k} \sum_{s=1}^{S_{jk}} \delta^s_{jk} u^s_{jk} \\
\text{subject to:}\ & \\
& Ax \le c \\
& u^0_{jk} + \sum_{s=1}^{S_{jk}} u^s_{jk} - x_{jk} = -\mathbb{E}[\xi_{jk}] \\
& u^0_{jk} \le d^1_{jk} - \mathbb{E}[\xi_{jk}] \\
& 0 \le u \le 1 \\
& x \in \mathbb{Z}_+ .
\end{aligned}
\tag{1}
$$

Notice that the decision variables of the linear integer program (1) are the vectors $x = (x_{jk})$, $u^0 := (u^0_{jk})$ and $u := (u^s_{jk})$, $s = 1, \ldots, S_{jk}$ (the vectors u and u^0 correspond to the slopes of the objective function of the second stage, which is piecewise linear). That is, problem (1) has $O(Snp)$ variables and $O(Snp + m)$ constraints (where $S := \max_{j,k} S_{jk}$), which is far smaller than the deterministic linear program corresponding to general two-stage programs with finitely many scenarios — in that case, it is well known that the number of constraints and variables is exponential on the number of scenarios, see for instance [5]. Thus, problem (1) can be solved by standard linear integer programming software.

It is worthwhile pointing out that, if one implements the booking policy based on seat allocations (we call it the *allocation policy* henceforth), then the objective function of [SLP] does correspond to the actual expected revenue resulting from a feasible policy x — though this is not true if the integrality constraint is relaxed. Note also that the solution obtained from [DLP] yields the same expected revenue as its rounded-down version. Moreover, it is easy to see that a rounded-down feasible solution to [DLP] is feasible for [SLP]. An immediate consequence of these facts is that the optimal allocation policy calculated from [SLP] is *never worse* than that of the DLP model in terms of expected total revenue. We emphasize that this is true in the present context of simple allocation policies, so such a conclusion may not hold for other policies.

2.1 Multi-stage models

We discuss now a multi-stage version of the SLP model described above, in which the policy is revised from time to time in order to take into account the information about demand learned so far. Suppose we divide the time horizon $[0, \tau]$ into $H+1$ stages numbered $0, 1, \ldots, H$. The stages correspond to some partition $0 = t_0 < t_1 < \ldots < t_{H-1} < t_H = \tau$ of the booking horizon, so that stage 0 corresponds to the beginning of the horizon and stage h ($h = 1, \ldots, H$) consists of time interval $(t_{h-1}, t_h]$. The decision variables at each stage are denoted $x^0, \ldots, x^H$, where $x^h = (x^h_{jk})$. Also, we associate with each stage h, $h = 1, \ldots, H$, random variables ξ^h_{jk} representing the total demand for itinerary-class (j, k) between stages $h-1$ and h, that is, $\xi^h_{jk} = N_{jk}(t_h) - N_{jk}(t_{h-1})$. We denote by ξ^h the random vector (ξ^h_{jk}). Notice that the decision vector x^h at stage h is actually a function of $x^0, x^1, \ldots, x^{h-1}$ and $\xi^1, \ldots, \xi^h$.

The resulting multi-stage model is written as follows:

$$\begin{aligned} &\max\; f^T x^0 + \mathbb{E}_{\xi^1}\left[Q_1(x^0, \xi^1)\right] \\ \text{subject to}\quad & \\ &Ax^0 \le c \\ &\;\; x^0 \in \mathbb{Z}_+. \end{aligned} \qquad \text{[MSSP]}$$

The function Q_1 is defined recursively as

$$\begin{aligned} Q_h(x^0, \ldots, x^{h-1}, \xi^1, \ldots, \xi^h) \;=\; & \max_{x^h}\; f^T x^h - f^T[x^{h-1} - \xi^h]^+ + \mathbb{E}_{\xi^{h+1}}\left[Q_{h+1}(x^0, \ldots, x^h, \xi^1, \ldots, \xi^{h+1})\right] \\ & \text{subject to} \\ & \quad Ax^h \le c - A\sum_{m=1}^{h} \min\{x^{m-1}, \xi^m\} \\ & \quad x^h \in \mathbb{Z}_+, \end{aligned}$$

($h = 1, \ldots, H-1$), where $[a]^+ := \max\{a, 0\}$ and the *max* and *min* operations are interpreted componentwise. Notice that we use the notation $\mathbb{E}_{\xi^{h+1}}\left[Q_{h+1}(x^0, \ldots, x^h, \xi^1, \ldots, \xi^{h+1})\right]$ to indicate the conditional expectation $\mathbb{E}\left[Q_{h+1}(x^0, \ldots, x^h, \xi^1, \ldots, \xi^{h+1}) \,|\, \xi^1, \ldots, \xi^h\right]$. For the final stage H we have

$$\begin{aligned} Q_H(x^0, \ldots, x^{H-1}, \xi^1, \ldots, \xi^H) \;&= -f^T x^H, \\ \text{where} \qquad x^H &= [x^{H-1} - \xi^H]^+. \end{aligned}$$

Observe that in the above formulations we use the equality $\mathbb{E}[\min\{a, Y\}] = a - \mathbb{E}[[a - Y]^+]$, where a is deterministic and Y is a random variable.

From the discussion above, we see that the multi-stage stochastic program [MSSP] is, in principle, a better model for the problem under study than [DLP] or [SLP], since it takes the stochastic and dynamic features of the problem into account. Several methods have been developed for multi-stage stochastic programs where all stages involve *linear* problems, see for instance [4, 6, 12]. One requirement for these algorithms to work is that the expected recourse function be *convex* (or concave, for a maximization problem). Moreover, these algorithms were devised for continuous problems — research on multi-stage integer problems is ongoing. Unfortunately, model [MSSP] described above does not fit that framework. In fact, as we show below, *even the continuous relaxation of that problem does not have a concave expected recourse function.*

Example 1. Consider a three stage programming problem corresponding to a single-leg, two-class problem, with the capacity equal to 4. There are two independent time periods. The demands for the two time periods are deterministic, respectively $\hat{\xi}^1 = (1, 2)$ and $\hat{\xi}^2 = (2, 2)$. The fares for each class are respectively $f_1 = \$500$ and $f_2 = \$100$. It is easy to check that $Q_1((1, 1), \hat{\xi}^1) = 1000$ and $Q_1((1, 3), \hat{\xi}^1) = 400$. However, the average of these values is $1400/2 = 700$, which is larger than 500, the value of Q_1 at the midpoint $(1, 2)$. Therefore, Q_1 is not concave.

It is worth noticing that the lack of concavity exists only for problems with three or more stages, the reason for that being the *min* operation in the constraint of the problem defining Q_1. Also, it is important to keep in mind that this issue arises only in the formulation we have presented. For example, DeMiguel and Mishra [10] circumvent that problem by proposing a model which is "partially non-anticipative", in the sense that the decision maker has perfect information about the *current* stage (but not about future ones); Möller et al. [16], on the other hand, use a different set of decision variables — cumulative allocations rather then per-stage ones — and obtain a multi-stage integer linear stochastic program, which without the integrality constraints yields concave functions (indeed, when the number of stages is two, the model in [16] is just [SLP]). Those models, however, can still be difficult to solve, especially for a large network with a large number of scenarios; to avoid that, in section 3 we will discuss an alternative approach to solve the allocation problem.

3 Re-solving SLP model

A natural alternative to the multi-stage approach described in section 2.1 is to revise the booking policy from time to time by *re-solving* a simpler model such as [SLP]. While intuitively it can be argued that such an approach may yield policies that are inferior to the ones given by [MSSP] — which by construction finds the optimal dynamic seat allocation policy —, we believe that the re-solving approach is worth studying, for several reasons. First, as discussed before, the solution of

[MSSP] is likely to be computationally intensive, especially for large networks. Re-solving a problem such as [SLP], on the other hand, is reasonably fast, since as described earlier each instance of [SLP] is equivalent to a moderately sized linear integer program. Moreover, it is clear that the complexity of [MSSP] increases rapidly as the number of stages grow, since the number of decision variables and constraints becomes larger; the complexity of re-solving, on the other hand, clearly grows linearly with the number of stages.

We formalize now the re-solving approach. As in the case of the multi-stage model, we partition the booking time horizon $[0, \tau]$ into segments $\{0\}, (0, t_1], (t_1, t_2], \ldots, (t_{H-1}, \tau]$ with $0 = t_0 < t_1 < \ldots < t_{H-1} < t_H = \tau$, and define $\xi^h, h \in \{1, \ldots, H\}$ as the demand during the time interval $(t_{h-1}, t_h]$. By $\hat{\xi}^h$ we denote a specific realization of the random variable ξ^h.

We initially (i.e. at time 0) solve the following problem:

$$\begin{aligned} \max\ & f^T \mathbb{E}\left[\min\left\{x, \sum_{m=1}^{H} \xi^m\right\}\right] \\ \text{subject to}\quad & \\ & Ax \le c \\ & x \in \mathbb{Z}_+. \end{aligned} \qquad \text{[SLPR-0]}$$

Let x^0 denote the optimal solution of the above problem. Then, at each time t_h, $h = 1, \ldots, H-1$ we solve

$$\begin{aligned} \max\ & f^T \mathbb{E}\left[\min\left\{x, \sum_{m=h+1}^{H} \xi^m\right\} \,\middle|\, \xi^1 = \hat{\xi}^1, \ldots, \xi^h = \hat{\xi}^h\right] \\ \text{subject to}\quad & \\ & Ax \le c - A \sum_{m=1}^{h} \min\{x^{m-1}, \hat{\xi}^m\} \\ & x \in \mathbb{Z}_+ \end{aligned} \qquad \text{[SLPR-}h\text{]}$$

and denote the optimal solution by x^h. Note that the above model makes use of the information up to time t_h — this is the reason why the constraints involve the realizations $\{\hat{\xi}^m\}$ instead of the random variables $\{\xi^m\}$. Notice also that the expectation in the objective function of [SLPR-h] is calculated with respect to the overall demand from period $h+1$ on. The idea is that this would be the problem one would solve if it was assumed that no more re-solving would occur in the future.

The idea of re-solving an optimization model to use the available information is not new. In the context of stochastic programming, this is sometimes called the *rolling forward* approach in the literature, see for instance [2] and [15]. The specific aspects of each problem, however, lead to different ways to implement the rolling mechanism. For example, in [2] a two-stage model is initially solved where the realizations at the second stage correspond to all possible scenarios of the original problem. That is, such a problem can be enormous and must be solved via some sampling method (see [18] for a compilation of results). In contrast, in our case the two-stage program [SLPR-0] deals only with the total demand $\sum_{m=1}^{H} \xi^m$. This reduces the number of scenarios drastically, especially when the distribution of $\sum_{m=1}^{H} \xi^m$ can be determined directly. Such is the case, for example, when the demand for each itinerary-class (j, k) arrives according to a Poisson process — in that case, $\sum_{m=1}^{H} \xi^m$ has Poisson distribution, which then can be truncated as discussed in section 2. This results in tractable two-stage simple recourse problems that can be solved exactly.

The approach of considering two-stage problems with increasingly small horizons has been used by Balasubramanian and Grossmann [1] in the context of a multi-period scheduling problem. They call it the *shrinking horizon framework*. Their motivation is similar to ours — to provide a practical scheme for a difficult multi-stage integer problem. Notice however that in our case we have the additional issue of lack of convexity, as discussed in section 2.1.

The idea of re-solving an optimization model to take advantage of the information accumulated so far has been a common practice in revenue management applications. As mentioned earlier, however, Cooper [7] shows a simple counter example where re-solving [DLP] leads to a *worse* policy (in terms of expected revenue) than what would be obtained if one had kept the original policy throughout. As we show next, this does not happen in case [SLP] is re-solved.

Proposition 1. *The allocation policy obtained from using models* [SLPR-h]*,* $h = 0, \dots, H-1$*, yields an expected revenue which is bigger or equal to that given by the allocation policy obtained from solving* [SLPR-0] *only.*

Proof. It suffices to show that the first re-solving cannot worsen the expected revenue. Let x^0 denote the optimal solution of [SLPR-0]. Then, during the time interval $(0, t_1]$, the booking allocation policy is x^0 regardless of whether we are re-solving or not. The difference happens after time t_1. For non-re-solving process, the policy to be used from time t_1^+ on is $x^1 := x^0 - \min\{x^0, \hat{\xi}^1\}$ because the non-re-solving process continues to apply the initial policy. The policy for the re-solving process is obtained by solving (SLPR-1), i.e.,

$$\max\ f^T \mathbb{E}\left[\min\left\{x, \sum_{m=2}^{H} \xi^m\right\} \,\middle|\, \xi^1 = \hat{\xi}^1\right]$$

subject to

$$Ax \leq c - A\min\{x^0, \hat{\xi}^1\}$$
$$x \in \mathbb{Z}_+.$$

Notice that, by definition of x^1,

$$Ax^1 \;=\; Ax^0 - A\min\{x^0, \hat{\xi}^1\} \;\leq\; c - A\min\{x^0, \hat{\xi}^1\}$$

for any possible realization of ξ_1. That is, x^1 is a feasible solution for the re-solving model, which means that the policy obtained by re-solving cannot be worse than x^1. Since the objective function of [SLPR-1] is the expected revenue from time t_1^+ on, it follows that re-solving cannot yield worse results than the initial policy. □

Proposition 1 shows that, in terms of expected revenue under the corresponding allocation policies, solving [SLPR-h] successively will keep improving (though perhaps not strictly) the booking policy. Notice that the re-solving method is somewhat similar to the multi-stage model in the sense that both yield dynamic booking policies that take the information available so far into account. As mentioned earlier, it is intuitive that in general [MSSP] gives a better solution. The proposition below formalizes that result.

Proposition 2. *Under the same partition setting, the allocation policy from multi-stage model* [MSSP] *yields an expected revenue which is bigger or equal to that given by the allocation policy obtained from solving* [SLPR-h] *successively.*

Proof. Let Ω_h be the set of all possible sample paths of $(\xi^1, \ldots, \xi^h)$. A feasible solution for problem [MSSP] has the form

$$x^0 \times \prod_{h=1}^{H} \prod_{\hat{\xi}^1, \ldots, \hat{\xi}^h \in \Omega_h} x^h(\hat{\xi}^1, \ldots, \hat{\xi}^h), \tag{2}$$

where $\prod$ indicates the cartesian product and each component $x^h(\hat{\xi}^1, \ldots, \hat{\xi}^h)$ satisfies

$$Ax^h(\hat{\xi}^1, \ldots, \hat{\xi}^h) \ \leq \ c - A \sum_{m=1}^{h} \min\{x^{m-1}(\hat{\xi}^1, \ldots, \hat{\xi}^{m-1}), \hat{\xi}^m\}. \tag{3}$$

On the other hand, consider model [SLPR-h] under a specific realization $\bar{\xi}^1, \ldots, \bar{\xi}^h$ in the scenario tree, and denote its optimal solution by $\bar{x}^h(\bar{\xi}^1, \ldots, \bar{\xi}^h)$. Consider the vector formed by the cartesian product of such solutions for all realizations and all stages, i.e.,

$$\bar{x}^0 \times \prod_{h=1}^{H} \prod_{\bar{\xi}^1, \ldots, \bar{\xi}^h \in \Omega_h} \bar{x}^h(\bar{\xi}^1, \ldots, \bar{\xi}^h),$$

It is clear that the resulting vector has the form (2). Moreover, since the [SLPR-h] problem has the constraints $Ax^h \leq c - A \sum_{m=1}^{h} \min\{x^{m-1}, \bar{\xi}^m\}$, it follows that $\bar{x}^h(\bar{\xi}^1, \ldots, \bar{\xi}^h)$ satisfies (3). Therefore, the combined solution from the [SLPR-h] models is actually a feasible solution in [MSSP]. □

It is important to notice that Proposition 2 is valid under the *same partition* into stages. Indeed, the flexibility of the re-solving approach allows for the inclusion of additional re-solving points without much burden — in other words, the complexity grows *linearly* with the number of stages, which in general is not true for the multi-stage model. It is natural then to compare the MSSP model and the re-solving SLP model with a refined partition. As the example below shows, including even one extra re-solving point may yield better results than solving the multi-stage model.

Example 2. Consider a single-leg problem with two independent booking classes, 1 and 2, with $f_1 = \$300,\ f_2 = \$200$. The capacity is equal to 15, and the booking time horizon has three time periods, $1, 2, 3$. During period 1, the distribution of demand for classes 1 and 2 is

$$\xi_1^1 = \begin{cases} 0 & \text{with probability } \frac{1}{2} \\ 1 & \text{with probability } \frac{1}{2} \end{cases}, \quad \xi_2^1 = 0 \text{ with probability one.}$$

Likewise, the distribution of demand during period 2 is

$$\xi_1^2 = \begin{cases} 3 & \text{with probability } \frac{1}{2} \\ 7 & \text{with probability } \frac{1}{2} \end{cases}, \quad \xi_2^2 = \begin{cases} 3 & \text{with probability } \frac{1}{2} \\ 5 & \text{with probability } \frac{1}{2} \end{cases}$$

and

$$\xi_1^3 = \begin{cases} 5 & \text{with probability } \frac{1}{2} \\ 7 & \text{with probability } \frac{1}{2} \end{cases}, \quad \xi_2^3 = \begin{cases} 4 & \text{with probability } \frac{1}{2} \\ 8 & \text{with probability } \frac{1}{2} \end{cases}$$

during period 3.

Because of the limited scale of this problem, the multi-stage model can be solved by enumeration. Suppose we solve a three-stage problem with the second and third stages defined respectively as time intervals $(0, 1]$ and $(1, 3]$. It is easy to check that the optimal solution from this model is $x^0 = (15, 0)^T$, $x_1^1 = (10, 5)^T$ (when $\xi_1^1 = 0$ happens) and $x_1^1 = (10, 4)^T$ (when $\xi_1^1 = 1$ happens). The expected total revenue is \$3900.

For the re-solving SLP approach, $x^0 = (9, 6)^T$ is the first stage decision. Although this solution does not coincide with that from the MSSP model, it turns out that, once we re-solve at time 1, we obtain the same expected revenue of \$3900 resulting from [MSSP]. When we include an extra re-solving point at time 2, the expected total revenue becomes \$4000, which is \$100, or 2.56% higher than that of MSSP model. For a large network, the improvement would be significant. This example suggests that applying the re-solving procedure can be more beneficial (in terms of expected revenue under the allocation policy) than solving a more complicated multi-stage model.

Another type of comparison between the MSSP model and its re-solving counterpart can be made when one has perfect information. The proposition and corollary below show that, in that case, re-solving is in fact optimal. Although the direct applicability of these results is limited (as the true problem is stochastic), the proofs illustrate the relationship between the two approaches.

Proposition 3. *Under perfect information, the policies given by models* [SLPR-0] *and* [MSSP] *are equivalent, in the sense that they yield the same expected revenue.*

Proof. Suppose there is perfect information, i.e. there is only one sample path, which we denote by $(\bar{\xi}^1, \ldots, \bar{\xi}^H)$. Each decision x^h is made with knowledge of the whole vector $(\bar{\xi}^1, \ldots, \bar{\xi}^H)$. Then,

[MSSP] is written as

$$\max \sum_{h=1}^{H} f^T \min\{x^{h-1}, \bar{\xi}^h\}$$

$$\text{subject to} \tag{4}$$

$$Ax^0 \leq c$$

$$Ax^1 \leq c - A\min\{x^0, \bar{\xi}^1\} \tag{5}$$

$$Ax^2 \leq c - A\min\{x^0, \bar{\xi}^1\} - A\min\{x^1, \bar{\xi}^2\} \tag{6}$$

$$\vdots$$

$$Ax^{H-1} \leq c - A\sum_{h=1}^{H-1} \min\{x^{h-1}, \bar{\xi}^h\} \tag{7}$$

$$x \in \mathbb{Z}_+.$$

It is clear from the above formulation that, if $\tilde{x} := (\tilde{x}^0, \ldots, \tilde{x}^{H-1})$ is an optimal solution for (4), then so is $(\min\{\tilde{x}^0, \bar{\xi}^1\}, \ldots, \min\{\tilde{x}^{H-1}, \bar{\xi}^H\})$. Since the region defined by each constraint (5)-(7) contains the region defined by the next inequality (recall that A has only non-negative entries), it follows that the above problem can be simplified to

$$\max f^T \sum_{h=1}^{H} x^{h-1}$$

$$\text{subject to} \tag{8}$$

$$A\sum_{h=1}^{H} x^{h-1} \leq c$$

$$x^{h-1} \leq \bar{\xi}^h \qquad h = 1, \ldots, H$$

$$x \in \mathbb{Z}_+.$$

Consider now the problem

$$\max f^T y$$

$$\text{subject to} \tag{9}$$

$$Ay \leq c$$

$$y \leq \sum_{h=1}^{H} \bar{\xi}^h$$

$$y \in \mathbb{Z}_+.$$

Notice that (9) is precisely problem [SLPR-0] under perfect information. Thus, to show the property stated in the proposition, it suffices to prove that the policies derived from (8) and (9) are equivalent.

To do so, define $C_y := \{x \text{ feasible in } (8) : \sum_{h=1}^{H} x^{h-1} = y\}$. Let F be a mapping from $\{C_y : y \in \mathbb{R}^{np}\}$ into the feasible region of (9), defined as $F(C_y) = y$. Consider now the mapping G that represents the application of the policy obtained from [SLPR-0]. We can express G as a mapping from the feasible region of (9) into $\mathbb{R}^{H\times np}$, defined as follows. For each pair (j,k), let $a_{jk} \le H$ be the largest number such that $y_{jk} \ge \sum_{h=1}^{a_{jk}} \bar{\xi}_{jk}^h$. Then, we define G as

$$G^{h-1}(y)_{jk} \;:=\; \begin{cases} \bar{\xi}_{jk}^h & 1 \le h \le a_{jk} \\ y_{jk} - \sum_{m=1}^{h-1} \bar{\xi}_{jk}^m & h = a_{jk}+1 \\ 0 & a_{jk}+2 \le h \le H. \end{cases}$$

Notice that from the above definition we have $0 \le G^{h-1}(y)_{jk} \le \bar{\xi}_{jk}^h$ for all $h = 1,\dots,H$. Moreover,

$$\sum_{h=1}^{H} G^{h-1}(y)_{jk} \;=\; \sum_{h=1}^{a_{jk}} \bar{\xi}_{jk}^h + y_{jk} - \sum_{m=1}^{a_{jk}} \bar{\xi}_{jk}^m + \sum_{h=a_{jk}+2}^{H} 0 \;=\; y_{jk}.$$

It follows that $G(y) := (G^0(y),\dots,G^{H-1}(y))$ is feasible for (8) and, in addition, $\sum_{h=1}^{H} G^{h-1}(y) = y$. That is, $G(y) \in C_y$. By viewing C_y as an equivalence class, we see that G is actually the inverse of F, i.e., F is a one-to-one mapping. Now, since F preserves the objective function value, we conclude that the policies from (8) and (9) are equivalent. $\square$

Propositions 1, 2 and 3 together yield the following result.

Corollary 1. *Under the same partition into stages and perfect information, the policies given by models* [SLPR-h], $h = 0,\dots,H-1$ *and* [MSSP] *are equivalent.*

4 Bid-price methods

The policies discussed in the previous sections are of allocation type — i.e., accept customers from a certain class until the corresponding allocations are used up. Another well-known type of policies is given by *bid prices*. In the context of airline booking, this means each leg has an incremental price. A booking request corresponds to seat occupation in one or more legs; the sum of the incremental prices for those legs is called bid price for this request. Then, the request is accepted only if its fare is bigger or equal to that amount. Notice that this method automatically provides a form of "nesting" even in a network environment, since by construction it cannot happen that a low-fare customer is accepted while a high-fare request is rejected.

A common way to determine bid prices is through the *dual variables* of the allocation problems discussed in section 2. Williamson [25] studies the case where the bid prices are the dual variables of [DLP]. This method is quite simple and easy to use. However, as pointed out in [21], it may behave poorly. A natural alternative is to look at the dual multipliers of [SLP]. A third method, proposed by Talluri and van Ryzin [21], calculates the dual multipliers of [SLP] under perfect information and averages the resulting values over a number of samples. Another approach is described in Higle and Sen [13], where the dual multipliers are calculated from a different stochastic programming problem (a leg-based seat allocation formulation). We refer to [20] for other approaches to compute bid prices.

It is clear from the structure of the bid-price policy that its form is too rigid — depending on the values of the bid prices, entire classes may be rejected. In practice, the bid prices are re-calculated on a regular basis in order to take into account new information about the demand, thus providing a more flexible policy. When the bid prices are obtained from dual multipliers of a mathematical program, this amounts to re-solving the problem with updated information, which is precisely the setting of section 3.

In light of the results of section 3, a natural question that arises is whether the expected revenue under a bid price policy can be guaranteed not to worsen with a re-solving approach. Unfortunately, the answer is negative, even if the bid prices are calculated from the SLP model. We show below a small example to illustrate this issue.

Example 3. Consider a one-leg model with two booking classes, the less price-sensitive customers (class 1) paying \$100 and more price-sensitive customers (class 2) paying \$60. The capacity is 4. The demands for those customers are denoted by ξ_1, ξ_2. Therefore the DLP model is

$$\max \{100x_1 + 60x_2 \,:\, x_1 + x_2 \le 4,\ 0 \le x_i \le \mathbb{E}[\xi_i]\}.$$

For an arbitrary time $t \le \tau$, let ξ_k^t denote the (random) number of arrivals of class k up to time t, and let $\hat{\eta}^t$ denote the actual number of sold seats up to time t. Therefore, the re-solving DLP model is

$$\max \{100x_1 + 60x_2 \,:\, x_1 + x_2 \le 4 - \hat{\eta}^t,\ 0 \le x_i \le \mathbb{E}[\xi_i - \xi_i^t]\}.$$

The tables below show the possible outcomes of the corresponding bid price policy, according to the values of the various quantities involved in the above problems.

Case 1 $\mathbb{E}[\xi_1] > 4$ and $\mathbb{E}[\xi_1 - \xi_1^t] > 4 - \hat{\eta}^t$:

Booking Methods	Acceptable classes through time t	Acceptable classes thereafter
DLP without re-solving	1	1
DLP with re-solving	1	1

Case 2 $\mathbb{E}[\xi_1] \le 4$ and $\mathbb{E}[\xi_1 - \xi_1^t] > 4 - \hat{\eta}^t$:

Booking Methods	Acceptable classes through time t	Acceptable classes thereafter
DLP without re-solving	1, 2	1, 2
DLP with re-solving	1, 2	1

Case 3 $\mathbb{E}[\xi_1] > 4$ and $\mathbb{E}[\xi_1 - \xi_1^t] \leq 4 - \hat{\eta}^t$:

Booking Methods	Acceptable classes through time t	Acceptable classes thereafter
DLP without re-solving	1	1
DLP with re-solving	1	1, 2

Case 4 $\mathbb{E}[\xi_1] \leq 4$ and $\mathbb{E}[\xi_1 - \xi_1^t] \leq 4 - \hat{\eta}^t$:

Booking Methods	Acceptable classes through time t	Acceptable classes thereafter
DLP without re-solving	1, 2	1, 2
DLP with re-solving	1, 2	1, 2

Suppose now the demand for the whole horizon has the following distribution:

$$\xi_1 = \begin{cases} 5 & \text{with probability } \frac{1}{2} \\ 1 & \text{with probability } \frac{1}{2} \end{cases}, \quad \xi_2 = \begin{cases} 2 & \text{with probability } \frac{1}{2} \\ 4 & \text{with probability } \frac{1}{2}. \end{cases}$$

Moreover, suppose that we can divide the time horizon into two periods such that in the first period there are two class-2 arrivals only. The capacity is $c = 4$. Notice that we have $\mathbb{E}[\xi_1] = 3 < 4$ and $\xi_1^t = 0$ w.p.1, so $\mathbb{E}[\xi_1 - \xi_1^t] = 3$.

From the tables above we see that the initial policy determined by the bid prices is to accept all the requests. Then, after the first period, two seats are occupied, i.e., $\hat{\eta}^t = 2$, and thus case 2 above applies. It follows that, when re-solving model [DLP] to obtain new bid prices, the policy becomes only accepting class-1 customers. It is not difficult to verify that the expected revenue for the second period is \$155.95 under non-re-solving policy, and \$150 under the re-solving one. Since the expected revenue for the first period is the same for both policies, it follows that the re-solving policy behaves worse than the non-resolving one.

Similar results are obtained for the case of bid prices generated from model [SLP] (under the same demand distribution as above), although the calculations are slightly more complicated. The solution of the SLP problem is $(2, 2)^T$, which implies that the incremental price for the leg is \$60. Thus, in the first time period two seats are allocated to the class-2 customers, so the remaining capacity for the second period is 2. When re-solving SLP model again, we get the new allocation at $(2, 0)^T$ which implies that the incremental price for the leg is \$100. That is, the re-solving SLP method changes policy from accepting all customers into accepting only class-1 customers. It follows that the expected revenue for the second period is the same as in the DLP case — \$155.95 for non-re-solving, \$150 for re-solving. This shows that re-solving can be worse under the bid-price policy, even if it is generated from the SLP model.

5 Choosing when to re-solve

An issue that does not seem to have been given attention in the literature is how to choose the times at which decisions are re-evaluated. The standard practice, both for multi-stage models as well as for re-solving approaches, appears to be to review the decisions at equally-spaced time points. However, there is no reason this must be done so, and one may benefit from a better choice of those times. In this section we discuss this issue in the context of the re-solving approach laid out in section 3.

To illustrate, consider a situation where re-solving is applied once, at some time t. That is, we have an initial allocation x^0 and a revised allocation x^1, which is obtained from the problem solved at time t. Using the notation defined earlier, let ξ^1 and ξ^2 be the vectors of total number of requests during intervals $(0, t]$ (stage 1) and $(t + 1, \tau]$ (stage 2) respectively. We shall write $\xi^1(t)$ and $\xi^2(t)$ to emphasize the dependence of these quantities on t, and similarly for x^1. Notice that, under this allocation policy, the expected revenue from time t on is given by $f^T\mathbb{E}\left[\min\{x^1(t), \xi^2(t)\}\right]$. Thus, the improvement from re-solving is given by

$$f^T\mathbb{E}\left[\min\{x^1(t), \xi^2(t)\}\right] - f^T\mathbb{E}\left[\min\{(x^0 - \min\{x^0, \xi^1(t)\}), \xi^2(t)\}\right] \geq 0. \tag{10}$$

The second term above discounts the revenue resulting from keeping policy x^0 from time t on. The term $\min\{x^0, \xi^1(t)\}$ gives the number of sold seats up to time t, so x^0 minus that quantity is the number of available seats at time t under the initial policy. Thus, in principle one could determine the value of t that maximizes the quantity in (10); but doing so, of course, is not practical, as it requires re-solving the model for each value of t is order to calculate $x^1(t)$.

Nevertheless, we can study a heuristic approach to determine appropriate re-solving points. We assume throughout this section that the arrival process of each itinerary-class is a (possibly non-homogeneous) Poisson process, so $\xi^1(t)$ and $\xi^2(t)$ are independent for any given t. Let us consider a simplified model which is restricted to one leg and where the random variables in the objective function are replaced with their expectations. At time 0 we solve the problem

$$\max\left\{f^T\min\{x, \mathbb{E}[\xi]\} \,:\, \sum_{k=1}^{n} x_k \leq c,\ x \geq 0\right\} \tag{11}$$

and the solution gives the initial allocation x^0. Assume that the classes are ordered such that $f_1 \geq f_2 \ldots \geq f_n$. Then, we can easily solve (11). The solution is

$$x^0 \;=\; \left(\mathbb{E}[\xi_1], \ldots, \mathbb{E}[\xi_q],\ c - \sum_{k=1}^{q}\mathbb{E}[\xi_k],\ 0, \ldots, 0\right),$$

where q is the smallest index such that $\sum_{k=1}^{q+1}\mathbb{E}[\xi_k] > c$. For simplicity, assume that $\sum_{k=1}^{q}\mathbb{E}[\xi_k] = c$, so $x^0_{q+1} = 0$.

Now consider the re-solving problem at time t. Let $\xi^1(t)$ and $\xi^2(t)$ be defined as before, so

$\xi^1(t) + \xi^2(t) = \xi$. Then, the re-solving problem is

$$\max f^T \min\{x, \mathbb{E}[\xi^2(t)]\}$$

$$\text{subject to} \quad \sum_{k=1}^{n} x_k \le c - \sum_{k=1}^{n} \min\{x_k^0, \hat{\xi}_k^1(t)\},$$

$$x \ge 0.$$

Note that the above problem is defined for a particular realization $\hat{\xi}_k^1(t)$ of $\xi_k^1(t)$. From the value of x^0 calculated above, we can rewrite the problem as

$$\begin{aligned} &\max f^T \min\{x, \mathbb{E}[\xi - \xi^1(t)]\} \\ \text{subject to} \quad &\sum_{k=1}^{n} x_k \le c - \sum_{k=1}^{q} \min\{\mathbb{E}[\xi_k], \hat{\xi}_k^1(t)\}, \\ &x \ge 0. \end{aligned} \tag{12}$$

Again this is easy to solve, and we get

$$x^1(t) \;=\; \left(\mathbb{E}[\xi_1 - \xi_1^1(t)], \ldots, \mathbb{E}[\xi_\ell - \xi_\ell^1(t)],\; c - \sum_{k=1}^{\ell} \mathbb{E}[\xi_k - \xi_k^1(t)],\; 0, \ldots, 0\right),$$

where ℓ is the smallest index such that

$$\sum_{k=1}^{\ell+1} \mathbb{E}[\xi_k - \xi_k^1(t)] \;>\; c - \sum_{k=1}^{q} \min\{\mathbb{E}[\xi_k], \hat{\xi}_k^1(t)\}.$$

Again, for simplicity let us discard the "residual", i.e., assume that $x_{\ell+1}^1(t) = 0$. Also, suppose that $\ell = q$ w.p.1. Then, the expected objective function value of problem (12) (calculated over the possible realizations of $\xi^1(t)$) is given by

$$\nu^1(t) \;=\; \mathbb{E}\left[f^T \min\{x^1(t), \mathbb{E}[\xi - \xi^1(t)]\}\right] \;=\; \sum_{k=1}^{q} f_k \mathbb{E}[\xi_k - \xi_k^1(t)].$$

Next, we calculate the value of re-solving. If we do not re-solve, the allocation at time t is given by $\bar{x}(t) = x^0 - \min\{x^0, \xi^1(t)\}$, which given the value of x^0 calculated above can be written as

$$\bar{x}(t) \;=\; \left(\left[\mathbb{E}[\xi_1] - \hat{\xi}_1^1(t)\right]^+, \ldots,\; \left[\mathbb{E}[\xi_q] - \hat{\xi}_q^1(t)\right]^+,\; 0, \ldots, 0\right).$$

The value of re-solving is given by the expected improvement in the objective value of problem (12) when using the optimal solution $x^1(t)$ instead of keeping $\bar{x}(t)$. The expected objective value at $\bar{x}(t)$ is given by

$$\bar{\nu}(t) \;=\; \mathbb{E}\left[f^T \min\{\bar{x}(t), \mathbb{E}[\xi - \xi^1(t)]\}\right] \;=\; \sum_{k=1}^{q} f_k \mathbb{E}\left[\min\left\{\left[\mathbb{E}[\xi_k] - \xi_k^1(t)\right]^+, \mathbb{E}[\xi_k - \xi_k^1(t)]\right\}\right].$$

In order to compare $\nu^1(t)$ and $\bar{\nu}(t)$, define $\Delta(t) := \nu^1(t) - \bar{\nu}(t)$. Then, we have

$$\Delta(t) \;=\; \sum_{k=1}^{q} f_k \left\{ \mathbb{E}[\xi_k - \xi_k^1(t)] - \mathbb{E}\left[\min\left\{ \left[\mathbb{E}[\xi_k] - \xi_k^1(t)\right]^+, \mathbb{E}[\xi_k - \xi_k^1(t)] \right\} \right] \right\} \tag{13}$$

To alleviate the notation, define $\mu_k := \mathbb{E}[\xi_k]$, and let $Z_k := \xi_k^1(t)$ (so $\lambda_k := \mathbb{E}Z_k \le \mu_k$). Note that the second term in (13) can be rewritten as

$$\begin{aligned} \mathbb{E}\left[\min\left\{[\mu_k - Z_k]^+, \mu_k - \lambda_k\right\}\right] \;&=\; \mathbb{E}\left[\min\left\{\mu_k - \min\{\mu_k, Z_k\}, \mu_k - \lambda_k\right\}\right] \\ &=\; \mu_k - \mathbb{E}\left[\max\left\{\min\{\mu_k, Z_k\}, \lambda_k\right\}\right], \end{aligned}$$

so by substituting the above into the expression for $\Delta(t)$ we have

$$\Delta(t) \;=\; \sum_{k=1}^{q} f_k \left(\mathbb{E}\left[\max\left\{\min\{\mu_k, Z_k\}, \lambda_k\right\}\right] - \lambda_k\right) \;=\; \sum_{k=1}^{q} f_k \mathbb{E}\left[[\min\{\mu_k, Z_k\} - \lambda_k]^+\right]. \tag{14}$$

Note that we can write the expectation inside the sum as

$$\begin{aligned} \mathbb{E}\left[[\min\{\mu_k, Z_k\} - \lambda_k]^+\right] \;&=\; \mathbb{E}\left[[\min\{\mu_k, Z_k\} - \lambda_k]^+ \mathbb{I}_{\{Z_k > \mu_k\}}\right] + \mathbb{E}\left[[\min\{\mu_k, Z_k\} - \lambda_k]^+ \mathbb{I}_{\{Z_k \le \mu_k\}}\right] \\ &=\; \mathbb{E}\left[(\mu_k - \lambda_k)\mathbb{I}_{\{Z_k > \mu_k\}}\right] + \mathbb{E}\left[[Z_k - \lambda_k]^+ \mathbb{I}_{\{\lambda_k \le Z_k \le \mu_k\}}\right] \\ &=\; (\mu_k - \lambda_k)P(Z_k > \mu_k) + \mathbb{E}\left[Z_k \mathbb{I}_{\{\lambda_k \le Z_k \le \mu_k\}}\right] - \lambda_k P\left(\lambda_k \le Z_k \le \mu_k\right). \end{aligned} \tag{15}$$

For simplicity, let us assume that μ_k and λ_k are integers. Then, since Z_k has Poisson distribution (with parameter $\lambda_k = \int_0^t \lambda_k(s)\, ds$, where $\lambda_k(\cdot)$ is the arrival rate function), it is easy to show that

$$\mathbb{E}\left[Z_k \mathbb{I}_{\{\lambda_k \le Z_k \le \mu_k\}}\right] \;=\; \lambda_k P\left(\lambda_k - 1 \le Z_k \le \mu_k - 1\right)$$

so in (14) we have

$$\Delta(t) \;=\; \sum_{k=1}^{q} f_k \left[(\mu_k - \lambda_k)P(Z_k > \mu_k) + \lambda_k \left(P(Z_k = \lambda_k - 1) - P(Z_k = \mu_k)\right)\right]. \tag{16}$$

Clearly, all terms on the right-hand side of the above equation can be evaluated numerically for a given value of t, so it is easy to find the re-solving point t that maximizes $\Delta(t)$. However, we are interested in developing a heuristics that allows for *multiple* re-solving points. To do so, we shall replace the probabilities in (16) with quantities that are easier to calculate.

Consider the term $P(Z_k > \mu_k)$. Using the Markov inequality $P(Z_k > \mu_k) \le \mathbb{E}Z_k/\mu_k$, we will replace that quantity with λ_k/μ_k, so the first term inside the brackets in (16) becomes simply $\lambda_k - \lambda_k^2/\mu_k$. Consider now the term $P(Z_k = \lambda_k - 1)$. Since Z_k is a Poisson random variable with mean λ_k, we have

$$P(Z_k = \lambda_k - 1) \;=\; \frac{e^{-\lambda_k}\lambda_k^{\lambda_k - 1}}{(\lambda_k - 1)!}.$$

Using Sterling's approximation $n! \approx n^n e^{-n}\sqrt{2\pi n}$ and also the approximation $[n/(n-1)]^n \approx e$, we have that

$$P(Z_k = \lambda_k - 1) \;\approx\; \frac{1}{\lambda_k}\sqrt{\frac{\lambda_k - 1}{2\pi}},$$

and so we see that the second term in (16) is less than or equal to $\sqrt{\frac{\lambda_k - 1}{2\pi}}$. It follows that the contribution of this term is small compared to $\lambda_k - \lambda_k^2/\mu_k$ and hence we will discard it. Next, suppose that we can actually compute different re-solving points for each class. Then, the problem of maximizing $\Delta(t)$ in (14) becomes separable and, together with the above approximations, it reduces to finding λ_k that maximizes $f_k(\lambda_k - \lambda_k^2/\mu_k)$. It is easy to check that the solution to this problem is $\lambda_k = \mu_k/2$. That is, we want to find t_k such that

$$\mathbb{E}[\xi_k^1(t_k)] \;=\; \mathbb{E}[\xi_k]/2. \tag{17}$$

This suggests that, for the original non-separable problem of maximizing $\Delta(t)$ in (16), we choose t such that

$$\sum_i f_k \mathbb{E}[\xi_k^1(t)] \;\approx\; \frac{1}{2}\sum_i f_k \mathbb{E}[\xi_k]. \tag{18}$$

We can give the following intuitive explanation for (18). From (10) we see that, if $\xi^2(t)$ is small with high probability, the improvement in revenue resulting from re-solving is minimal. That is, we want to pick a re-solving point t in such a way that the demand from time t on (i.e. $\xi^2(t)$) is "high enough". This suggests taking t not too close to τ. On the other hand, if t is too close to zero, then $\xi^1(t)$ is small and x^0 is close to x^1, so again from (10) we see that the improvement is minimal. So, it seems sensible to choose t such that $\mathbb{E}[\xi^1(t)] \approx \mathbb{E}[\xi^2(t)]$, i.e., $\mathbb{E}[\xi^1(t)] \approx \mathbb{E}[\xi]/2$. We then weight the terms by the corresponding fares to take revenue into account.

The above heuristics can be generalized to multiple re-solving points. Suppose we decide we want to re-solve R times. Then the time we pick for the r^{th} re-solving point is t such that

$$\sum_k f_k \mathbb{E}[\xi_k(t)] \;\approx\; \frac{r}{R+1}\sum_k f_k \mathbb{E}[\xi_k]. \tag{19}$$

We omit the superscript 1 from $\xi_k(t)$, which represents class-k demand up to time t. Notice that $\xi_k = \xi_k(\tau)$. One way to pick t such that (19) holds is to solve the one-dimensional problem $\min_t |\sum_k f_k \mathbb{E}[\xi_k(t)] - r/(R+1)\sum_k f_k \mathbb{E}[\xi_k]|$.

The above procedure is intuitive in the single leg case, since there is a natural ordering of the classes by fare. However, as mentioned earlier the situation is more complicated when dealing with a network environment. For example, consider a situation where the fare for a certain itinerary-class pair that uses legs 1 and 2 is \$130, whereas the fare for another itinerary-class pair that uses leg 1 only is \$100, and the fare for another itinerary-class pair that uses leg 2 only is \$70. Intuitively, if one expects to see more arrivals of the latter classes, then the second class should be preferred over the first one when making decisions about leg 1. One way to quantify this is through the *net contribution* of each class. Suppose for example that the bid price associated with each leg is \$40. Then, the net contribution of the first itinerary-class is \$130 − \$80 = \$50, whereas the net contribution of the second one is \$100 − \$40 = \$60. That is, the second class is more profitable even though its fare is smaller.

It is clear that, in a single-leg environment, the net contribution is actually the fare level. For networks, one can apply heuristic procedures to rank the classes based on the net contribution. Such procedures are proposed in [3] and [9], for example. Borrowing from their ideas, we apply the follow steps to aggregate the demand vector into a one-dimensional quantity. Let f_{jk} be the fare level for certain itinerary-class pair (j, k) and let S_{jk} denote the set of legs which are used for that itinerary. We use the following algorithm to determine R re-solving points:

Algorithm 1

1. Solve model [SLP] at time 0. Let p_i denote the bid price for leg i, obtained from the dual variables of the continuous relaxation of [SLP].

2. Let q_{jk} be the net contribution of itinerary-class (j, k), calculated as $f_{jk} - \sum_{i \in S_{jk}} p_i$.

3. For each r, $r = 1, \ldots, R$, the r^{th} re-solving point t_r is a t that satisfies

$$\sum_{j,k} q_{jk} \mathbb{E}[\xi_{jk}(t)] \;\approx\; \frac{r}{R+1} \sum_{j,k} q_{jk} \mathbb{E}[\xi_{jk}],$$

   where as before $\xi_{jk}(t)$ is the class-k demand for itinerary j up to time t.

It is worth pointing out that, when the arrival processes $\{\xi_{jk}(t)\}$ are *homogeneous* Poisson processes, we have $\mathbb{E}[\xi_{jk}(t)] = \lambda_{jk} t$ for all t and hence the above algorithm yields $t_r := r\tau/(R+1)$, i.e., equally-spaced time points — which seems reasonable given the homogeneity of the arrivals. As we shall see in the next section, however, in the absence of homogeneity the choice of re-solving times becomes very important.

6 Numerical results

In this section we describe the results from numerical experiments performed with the policies discussed above. Although our data set was randomly generated, we tried to mimic real data as much as possible. To do so we imposed the following features, which according to Weatherford et al. [24] are characteristic of actual booking processes. They are (1) uncertain number of potential customers; (2) uncertain mix of high- and low-fare customers; (3) uncertain order of arrivals; and (4) high-fare customers tend to arrive *after* the low-fare ones.

The first example is a 10-leg network described in Figure 1 below. We consider all flights to/from the hub from/to each city, as well as the flights between two cities connecting at the hub.

Therefore, there are 30 possible itineraries in the network. There are two booking classes for each flight, with the proportion of 1:3 between high and low fare classes in terms of total requests. Following [24], we model the booking process by a doubly stochastic non-homogeneous Poisson

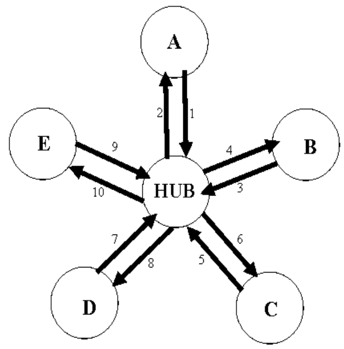

Figure 1: Example 1

process (NHPP), where the arrival intensity at time t has gamma distribution. More specifically, for each itinerary j let $\lambda_{j1}(t)$ and $\lambda_{j2}(t)$ be the arrival intensity of respectively high-fare and low-fare customers at time t. Denote by $\alpha_j > 0$ the total expected number of requests for itinerary j over the booking horizon (i.e., for both classes together). Let G_j be a random variable with gamma distribution with shape parameter α_j and scale parameter $\beta' = 1$ (that is, the density function of G_j is $f_j(x) = \frac{(x/\beta')^{\alpha_j - 1} e^{-x/\beta'}}{\beta' \Gamma(\alpha_j)}$, $x \geq 0$).

We define $\lambda_{jk}(t)$, $k = 1, 2$, as

$$\begin{aligned} &\lambda_{jk}(t) = \beta_{jk}(t) \times G_j \times \psi_k \\ \text{where} \qquad &\beta_{jk}(t) = \frac{1}{\tau}\left(\frac{t}{\tau}\right)^{a_{jk}-1}\left(1 - \frac{t}{\tau}\right)^{b_{jk}-1} \frac{\Gamma(a_{jk} + b_{jk})}{\Gamma(a_{jk})\Gamma(b_{jk})}. \end{aligned}$$

The parameters ψ_1, ψ_2 are set with the goal of reflecting the proportion of arrivals for high- and low-fare customers. We take this proportion to be 1:3 in all itineraries, so we set $\psi_1 = 0.25$ and $\psi_2 = 0.75$. Notice that, for each t, $\lambda_{jk}(t)$ has gamma distribution with shape parameter α_j and scale parameter $\beta_{jk}(t)\psi_k$. In particular, $\mathbb{E}[\lambda_{jk}(t)] = \alpha_j \beta_{jk}(t)\psi_k$ and hence the total expected number of arrivals for itinerary j is $\int_0^\tau \mathbb{E}[\lambda_{j1}(t)] + \mathbb{E}[\lambda_{j2}(t)] dt = \alpha_j\psi_1 + \alpha_j\psi_2 = \alpha_j$, which is consistent with our definition of α_j.

The parameters $\beta_{jk}(t)$ are selected to reflect the arrival patterns of different classes. High-fare customers tend to arrive close to the end of the booking horizon, whereas low-fare customers usually appear early in the booking process. To model that, we set $a_{j1} > b_{j1}$ (high-fare customers), and $a_{j2} < b_{j2}$ (low-fare customers). In our example we used $a_{jk}, b_{jk} \in \{2, 6\}$ for all j, k.

From Figure 1 we see that there are 10 one-leg and 20 two-leg itineraries. For two-leg itineraries, we set the total expected number of requests equal to 100, that is, $\alpha_j = 100$. The high and low fares are respectively $f_{j1} = \$500$ and $f_{j2} = \$100$. For one-leg itineraries, we set the total expected number of requests equal to 40, that is, $\alpha_j = 40$, with the high and low fares set as $f_{j1} = \$300$ and $f_{j2} = \$80$. All legs in the network have capacity equal to 400, and the booking horizon has length $\tau = 1000$ time units.

The second example is depicted in Figure 2. Again, we consider two classes for each itinerary. Notice that there are 10 one-leg, 12 two-leg, and 8 three-leg itineraries. The expected number of requests are 60, 150, and 100 respectively. The fare levels for different type of itineraries are set as (\$300, \$80), (\$500, \$100), and (\$700, \$200). The parameters a_{jk}, b_{jk}, ψ_1 and ψ_2 are the same as in the first example, as well as the horizon length. The leg capacities are 400 for the arcs connecting the hubs with the satellite nodes, and 1,000 for the arcs connecting the two hubs.

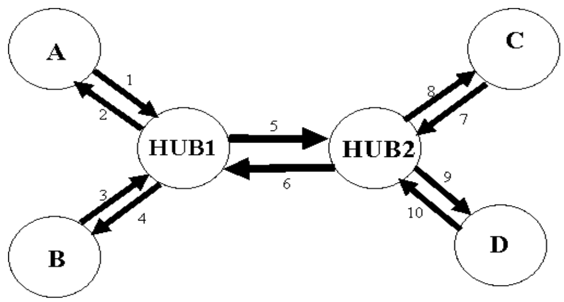

Figure 2: Example 2

For each of the problems we implemented four basic policies: DLP, SLP, DLP-based bid price and SLP-based bid price. The linear (integer) programs required to determine the allocations were solved using the software package XPressMP^{TM} from Dash Optimization (under the Academic Partnership Program). For each of the policies, we considered the effect of solving it only once as well as twice and five times over the booking horizon. The re-solving points were calculated using two methods — the standard approach of equally-spaced points as well as Algorithm 1 from section 5.

For these examples, finding the appropriate re-solving points according to Algorithm 1 was very simple, since the demand distributions used in the examples allow us to calculate $\mathbb{E}[\xi_{jk}(t)]$ exactly. It turns out that $\mathbb{E}[\xi_{jk}(t)] = \alpha_j \psi_k B_{jk}(t/\tau)$, where

$$B_{j1}(s) := -6s^7 + 7s^6$$
$$B_{j2}(s) := -6s^7 + 35s^6 - 84s^5 + 105s^4 - 70s^3 + 21s^2$$

for all j. Notice that the expected number of arrivals of itinerary-class (j, k) over the whole horizon is $\mathbb{E}[\xi_{jk}] = \mathbb{E}[\xi_{jk}(\tau)] = \alpha_j \psi_k B_{jk}(1) = \alpha_j \psi_k$. Figure 3 shows the plot of the function $H(t) := \sum_{j,k} q_{jk} \mathbb{E}[\xi_{jk}(t)]$ for both examples. It is clear that re-solving occurs more often as the slope of H gets larger; thus, equally-sized intervals are appropriate only when H is linear (which, as mentioned earlier, happens when the arrival process is a homogeneous Poisson), but this is not the case in these examples. The algorithm suggests re-solving more often as the end of the horizon approaches, which reflects the fact that high-fare customers tend to book later than low-fare ones.

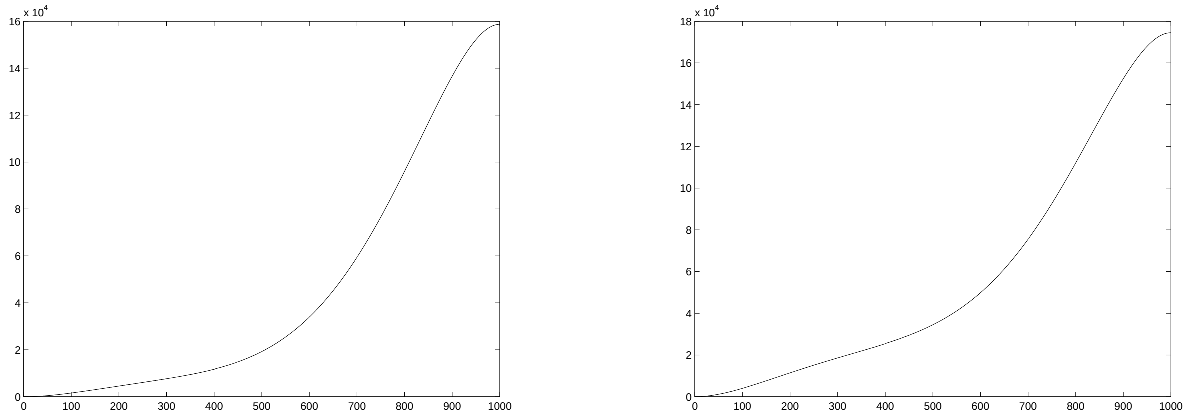

Figure 3: Graph of the function $H(t) = \sum_{j,k} q_{jk}\mathbb{E}[\xi_{jk}(t)]$ for Examples 1 (left) and 2 (right).

The results in Tables 1-2 show the average revenue for each policy and each example. These numbers were obtained by building a simulation model whereby we simulate the arrival process and apply the corresponding booking policies. The accrued revenue is the sum of the fares of accepted customers, and we compute the average revenue over 1000 replications. Next to each number we display the half-width of a 95% confidence interval for the expected revenue obtained with the corresponding allocation policy. In order to facilitate the comparison, the same streams of random numbers were used in each run, so that all methods see the same arrivals (each replication, of course, uses a different stream). The re-solving times listed on the third an fifth rows of the tables were obtained with Algorithm 1.

	DLP–alloc.	SLP–alloc.	DLP–bid	SLP–bid
No re-solving	401980 ± 406	415410 ± 598	347690 ± 967	347990 ± 932
Re-solve at $t = 500$	404520 ± 426	418012 ± 587	379562 ± 915	373760 ± 941
Re-solve at $t = 757$	407800 ± 433	419298 ± 566	347988 ± 924	348430 ± 911
Re-solve at $t = 200, 400, 600, 800$	409740 ± 468	419262 ± 594	391768 ± 913	414310 ± 894
Re-solve at $t = 587, 712, 797, 875$	409138 ± 464	421894 ± 613	360270 ± 897	359900 ± 923

Table 1: Expected revenue under allocation and bid-price policies, Example 1.

The results confirm the findings reported in sections 3 and 4 that the allocation policy based on model [SLP] is robust, in the sense that re-solving cannot worsen the revenue. Although the numbers suggest that the same is true for bid prices or DLP-based allocation, we know this is not necessarily the case, as discussed in sections 3 and 4. Moreover, we see the effect of choosing the re-solving points more carefully, as the revenue under the allocation policy goes up in that case.

	DLP–alloc.	SLP–alloc.	DLP–bid	SLP–bid
No re-solving	583630 ± 552	595620 ± 726	518470 ± 984	520210 ± 938
Re-solve at $t = 500$	584610 ± 560	597258 ± 698	554080 ± 862	557150 ± 856
Re-solve at $t = 735$	588015 ± 614	601156 ± 746	522920 ± 1003	525840 ± 958
Re-solve at $t = 200, 400, 600, 800$	592387 ± 554	603088 ± 606	578758 ± 987	602890 ± 894
Re-solve at $t = 504, 682, 782, 867$	594021 ± 536	604859 ± 598	555598 ± 1078	561464 ± 1022

Table 2: Expected revenue under allocation and bid-price policies, Example 2.

For bid-price policies, however, the re-solving times obtained with Algorithm 1 were less effective than equally-spaced points. This is not entirely surprising, since the rationale behind Algorithm 1 described in section 5 is based on improving revenues under the allocation policy.

Finally, in order to assess the quality of these results we computed the *wait-and-see* solutions. These solutions determine the actions that would be taken if all uncertainty was known in advance. Although this does not yield a practical policy, it does provide an upper bound for the expected revenue. Table 3 displays the results obtained from sampling scenarios and computing 95% confidence intervals for the wait-and-see value. A comparison of these values with the ones in Tables 1-2 shows that the best policies in those examples (SLP allocation with re-solving times from Algorithm 1) yielded revenues of about 97% of the upper bounds, which suggests that those policies may perform well in practice.

	Wait-and-see value
Example 1	432730 ± 593
Example 2	623530 ± 706

Table 3: Wait-and-see values for examples 1 and 2

7 Conclusions

We have discussed the airline booking process based on the origin-destination model. More specifically, we have presented a multi-stage stochastic programming formulation to the seat allocation problem, which extends the traditional two-stage model proposed in the literature. Our study suggests that solving this multi-stage problem exactly may be difficult, because of the lack of convexity properties. In order to circumvent that obstacle, we have used an approximation based on solving a sequence of two-stage stochastic linear integer programs (SLPs) with simple recourse.

Our analysis suggests that the proposed approach is robust, in the sense that solving successive SLPs can only improve the expected revenue — i.e., it is never better not to re-solve. While intuitive, to our knowledge such a property had not been shown in the literature. Moreover, this gives an advantage of SLPs over the standard deterministic linear program formulation, for which it is known that re-solving can actually "backfire". As it turns out, the same phenomenon happens with some bid-price policies, and we have presented some examples where re-solving worsens the expected revenue.

We have also shown that, under perfect information, the multi-stage model and the re-solving approach for the seat allocation problem coincide; this may have some algorithmic implications (e.g., by applying the progressive hedging algorithm of Rockafellar and Wets [17]), which is a topic for further research. Note that theoretical comparisons involving bid-prices policies are harder to obtain, since the revenue accrued with those policies highly depends on the arriving order.

The flexibility of the re-solving approach has allowed us to propose a heuristic method whereby the re-solving points are chosen with the goal of maximizing the incremental expected revenue; our numerical results, run for two relatively small-sized models (each with a different network structure) suggests that the approach is effective.

We must remark that we have not included in our models some of the recent developments proposed in the literature, such as algorithms that include nesting of classes (see, e.g., [3, 23]) and consumer-choice modeling techniques (see, e.g., [11, 22]). There are two basic reasons for our decision: first, incorporation of the above features leads to different models which lie outside of the scope of this paper — for example, the problems in [3, 23] are non-convex problems that are solved with simulation-based methods, whereas the linear programs in [11, 22] are of different nature than the ones discussed here. Second, our conversations with people in the airline industry have shown to us that the basic origin-destination model, particularly the deterministic linear programming formulation, is widely used in practice; thus, our goal is to provide the practitioners an easily implementable algorithm that can improve upon what is currently in use. Again, our theoretical and numerical results show that the methods proposed in this paper have the potential to accomplish that goal. Nevertheless, we plan to study the applications of those methods under other settings, particularly when consumer choice is incorporated into the model. Research on that topic is underway.

References

[1] J. Balasubramanian and I. Grossmann. Approximation to multistage stochastic optimization in multiperiod batch plant scheduling under demand uncertainty. *Ind. Eng. Chem. Res.*, 43: 3695–3713, 2004.

[2] M. Bertocchi, V. Moriggia, and J. Dupacova. Horizon and stages in applications of stochastic programming in finance. *Annals of Operations Research*, 142(1):63–78, 2006.

[3] D. Bertsimas and S. De Boer. Simulation-based booking limits for airline revenue management. *Operations Research*, 53(1):90–106, 2005.

[4] J. R. Birge. Decomposition and partitioning methods for multistage stochastic linear programs. *Operations Research*, 33(5):989–1007, 1985.

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

[6] J. R. Birge, C. J. Donohue, D. F. Holmes, and O. G. Svintsitski. A parallel implementation of the nested decomposition algorithm for multistage stochastic linear programs. *Mathematical Programming*, 75:327–352, 1996.

[7] W. L. Cooper. Asymptotic behavior of an allocation policy for revenue management. *Operations Research*, 50:720–727, 2002.

[8] W. L. Cooper and T. Homem-de-Mello. A class of hybrid approaches for revenue management. Working paper, University of Minnesota, Department of Mechanical Engineering. Available at `http://www.menet.umn.edu/~billcoop/abstracts.html#RMSO`, 2003.

[9] S. V. de Boer, R. Freling, and N. Piersma. Mathematical programming for network revenue management revisited. *European Journal of Operational Research*, 137:72–92, 2002.

[10] V. DeMiguel and N. Mishra. A multistage stochastic programming approach to network revenue management. Working paper, London Business School, 2006.

[11] G. Gallego, G. Iyengar, R. Phillips, and A. Dubey. Managing flexible products on a network. CORC Technical Report Tr-2004-01, IEOR Department, Columbia University, 2004.

[12] H. I. Gassmann. MSLiP: A computer code for the multistage stochastic linear programming problem. *Mathematical Programming*, 47:407–423, 1990.

[13] J. L. Higle and S. Sen. A stochastic programming model for network resource utilization in the presence of multi-class demand uncertainty. In W. T. Ziemba and S. W. Wallace, editors, *Applications of Stochastic Programming.* SIAM Series on Optimization, 2006. Forthcoming.

[14] P. Kall and S. W. Wallace. *Stochastic Programming.* John Wiley & Sons, Chichester, England, 1994.

[15] M. I. Kusy and W. T. Ziemba. A bank asset and liability management model. *Operations Research*, 34:356–376, 1986.

[16] A. Möller, W. Römisch, and K. Weber. A new approach to O&D revenue management based on scenario trees. *Journal of Revenue and Pricing Management*, 3(3):265–276, 2004.

[17] R. T. Rockafellar and R. J.-B. Wets. Scenarios and policy aggregation in optimization under uncertainty. *Mathematics of Operations Research*, 16(1):119–147, 1991.

[18] A. Shapiro. Monte Carlo sampling methods. In A. Ruszczynski and A. Shapiro., editors, *Handbook of Stochastic Optimization*. Elsevier Science Publishers B.V., Amsterdam, Netherlands, 2003.

[19] J. Subramanian, S. Stidham, and C. J. Lautenbacher. Airline yield management with overbooking, cancellations, and no-shows. *Transportation Science*, 33:147–168, 1999.

[20] K. Talluri and G. Van Ryzin. *The Theory and Practice of Revenue Management*. Kluwer Academic Publishers, Dordrecht, Netherlands, 2004.

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

[22] G. van Ryzin and Q. Liu. On the choice-based linear programming model for network revenue management. Working paper, Columbia Business School, 2004.

[23] G. van Ryzin and G. Vulcano. Simulation-based optimization of virtual nesting controls for network revenue management. Working paper DRO-2003-01, Columbia University, Graduate School of Business, 2003.

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

[25] E. L. Williamson. *Airline Network Seat Control*. PhD thesis, Massachusetts Institute of Technology, 1992.