An Integrated Approach to Single-Leg Airline Revenue Management: The Role of Robust Optimization

Ş. İlker Birbil* J.B.G. Frenk† Joaquim A.S. Gromicho‡ Shuzhong Zhang§

Econometric Institute Report E.I. 2006-20

ABSTRACT. In this paper we introduce robust versions of the classical static and dynamic single leg seat allocation models as analyzed by Wollmer, and Lautenbacher and Stidham, respectively. These robust models take into account the inaccurate estimates of the underlying probability distributions. As observed by simulation experiments it turns out that for these robust versions the variability compared to their classical counter parts is considerably reduced with a negligible decrease of average revenue.

Keywords: airline revenue management; single-leg problems; static models; dynamic models; robust optimization

1 Introduction

Airline seat allocation problems on single legs or networks play a prominent role within the revenue management literature. This field expanded rapidly in recent years and for an overview on revenue management up to 1999 we refer the reader to [11], while developments occurring after this work are discussed in the recent book by Talluri and Ryzin [15]. Although many practical seat allocation problems observed in the airline industry are network based, single leg seat allocation problems still play an important role. This is mainly due to two reasons: Firstly, in general the network based airline seat allocation problems are extremely difficult to solve. Therefore, different heuristics, which required the solution of many single leg problems, were developed. Secondly, some small airline companies, like charter flight companies commonly seen in Europe, have special one-hub networks with single legs. Therefore for those companies managing their seat allocation over the network requires solving only single leg problems. Among the single leg problems, one may distinguish static

*Faculty of Engineering and Natural Sciences, Sabanci University, Orhanli-Tuzla, 34956, Istanbul, Turkey.
E-mail: sibirbil@sabanciuniv.edu
†Econometrics Institute, Erasmus University Rotterdam, Postbus 1738, 3000 DR Rotterdam, The Netherlands.
E-mail: frenk@few.eur.nl
‡Vrije Universiteit, Amsterdam, The Netherlands & ORTEC International, Gouda, The Netherlands.
E-mail: jgromicho@feweb.vu.nl or jgromicho@ortec.nl
§Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, Shatin, Hong Kong. Research supported by Hong Kong RGC Earmarked Grants CUHK4242/04E and CUHK418505.
E-mail: zhang@se.cuhk.edu.hk

and dynamic models. Actually, the static models can be further categorized into two types. The first type assumes that only the distribution of the demand for the different fare classes is known. Since the objective is to maximize the expected revenue, this leads to the formulation of mathematical programming models. Examples of such models are given in [17, 5, 15]. The second type assumes that the demands for different fare classes arrive in non overlapping time periods in the order of increasing fare class prices. Given a realization of a particular fare class demand, one needs to decide how much of this demand is allocated to seats, under the probabilistic information on the demand for the remaining higher priced fare classes. This model can be solved by dynamic programming, where the stages correspond to fare classes. Examples of such models under different assumptions are presented for two fare classes in [10, 12], and for more than two fare classes in [1] (a heuristic approach generalizing the rule of Littlewood) and also in [18, 3, 13]. Finally, dynamic single leg models take into account the actual order of arrival of different fare class customers and so the decision to accept or reject a specific fare class customer is not static, but may change over time. In this case stages correspond to time periods. Examples of such models under different assumptions are given in [9, 8, 16].

In this paper we first review, in the section on static models, the mathematical programming formulation of the static single leg problem already given by Wollmer [17] in a more complicated network environment (see also [5, 15]). However, in these references only a binary linear programming formulation is given without any special purpose algorithms to solve those formulations. For the more special single leg case considered here, we give in Section 1 a fast special purpose algorithm to solve this model. Moreover, we present also in Section 1 a new robust formulation of the mathematical programming model, which takes into account the inaccurate estimate of the probability distributions of the total demand for the different fare classes. As shown in Section 5 it will turn out in our simulation experiments that the variability of the realized revenues is considerably smaller for the robust version. At the same time due to the conservative behavior of the robust model, the average revenues for the classical single static model are slightly higher. In Section 3 we then review the standard classical dynamic single leg problem as discussed in [8] and propose, also for this model, a new robust version. This robust version takes again into account the inaccurate estimates of the probabilities of the arrival process. In the same section we also propose, for the classical dynamic model, an extension to batch arrivals in each period. Again from our simulation results in section 5 we observe the same behavior as observed for the static models. In Section 4 we consider shortly which model we have to use in case of perfect information. Then we compare the three different models (static, dynamic and complete information) extensively by means of simulation in Section 5. Our simulation results show that that the cost of having incomplete information is relatively small. Finally, in Section 6 we conclude the paper.

We adopt a standard notation in our paper. The difference between the vectors and scalars should be clear from the text. The boldface letters are used to denote the random variables.

2 Static Models

In this section we are interested in the optimal allocation of the seat capacity C on a given flight among the m different fare classes. If the demand d_i for each fare class i, $1 \leq i \leq m$, is known in advance, it is trivial to solve this allocation problem which can be modelled in the following way. Let x_i denote the number of reserved seats for fare class i at the beginning of the booking period. We assume that fare class i customers do not consider the possibility of buying a ticket from a different fare class. Thus, once no fare class i ticket is available, then it follows that $\min\{x_i, d_i\}$ will be the number of occupied fare class i seats on the selected flight. To determine the optimal allocation of the different fare classes over the given capacity C, we need, therefore, to solve the following optimization problem

$$\begin{array}{rll} v_1(C) := & \max & \sum_{i=1}^m r_i \min\{x_i, d_i\} \\ & \text{s.t.} & \sum_{i=1}^m x_i \leq C, \\ & & x \in \mathbb{Z}_+^m, \end{array} \tag{2.1}$$

where r_i denotes the price of a fare class i seat. In case $r_1 < r_2 < \ldots < r_m$, it is obvious that an optimal allocation is given as follows: Consider demand d_i and price r_i for each fare class i, and assign all the seats to the higher-priced customers as long as the capacity is still available. To formalize the algorithm, introduce $S_n := \sum_{j=n}^m d_j$ with $d_0 := 0$ and $N(C) = \min\{0 \leq n \leq m \mid S_n \leq C\}$. Then, the optimal solution of optimization problem (2.1) is given by

$$x_i^* = \begin{cases} d_i, & \text{if } i \geq N(C) \\ C - S_{N(C)}, & \text{if } i = N(C) - 1 \\ 0, & \text{if } i < N(C) - 1. \end{cases} \tag{2.2}$$

The associated optimal objective function value as a function of the capacity C is given by

$$v_1(C) = \sum_{i=N(C)}^{m} r_i d_i + \left(C - S_{N(C)}\right) r_{N(C)-1},$$

which is, clearly, a piecewise linear concave function.

However, usually the demand for fare class i is a random variable $\mathbf{D}_i$ and we do not know in advance its realization. We may, however, estimate the distribution of the demand. Let $\mathbf{D}_i(\omega)$ be a realization of the demand $\mathbf{D}_i$ and x_i be the number of reserved seats for fare class i. Consequently, the total revenue is given by $\sum_{i=1}^m r_i \min\{x_i, \mathbf{D}_i(\omega)\}$. This shows that the expected revenue equals $\sum_{i=1}^m r_i \mathbb{E}\left(\min\{x_i, \mathbf{D}_i\}\right)$, and so, our *static* decision model for random demand is given by

$$\begin{array}{rll} v_2(C) := & \max & \sum_{i=1}^m r_i \mathbb{E}\left(\min\{x_i, \mathbf{D}_i\}\right) \\ & \text{s.t.} & \sum_{i=1}^m x_i \leq C, \\ & & x \in \mathbb{Z}_+^m. \end{array} \tag{2.3}$$

This static model was first formulated by Wollmer [17] in a much more complicated network environment and became a classical model in this field. Since the simpler single-leg version is a standard separable problem, it can be solved by dynamic programming. Introduce for every $p \leq m$

and $y \in \{0, ..., C\}$ the value $R_p(y)$ as the maximal expected revenue for fare classes p up to m if at most capacity y is reserved for those fare classes, i.e.,

$$R_p(y) = \max\left\{\sum_{i=p}^{m} r_i \mathbb{E}(\min\{x_i, \mathbf{D}_i\}) \mid \sum_{i=p}^{m} x_i \leq y, x_i \in \mathbb{Z}, i = p, \cdots, m\right\}.$$

By the optimality principle of Bellman it now follows for every $y \in \{0, \cdots, C\}$ and $p + 1 \leq m$ that

$$R_p(y) = \max_{0 \leq x_p \leq y} \{R_{p+1}(y - x_p) + r_p \mathbb{E}(\min\{x_p, \mathbf{D}_p\})\}.$$

Since clearly $R_m(y) = r_m \mathbb{E}(\min\{y, \mathbf{D}_m\})$, $y \in \{0, 1, ..., C\}$, we can recursively compute the optimal objective value $R_1(C)$. The computational complexity of this dynamic programming approach is of the order of $O(mC^2)$.

Clearly, to apply this approach we need an efficient algorithm to compute the function values $\mathbb{E}(\min\{x_i, \mathbf{D}_i\})$. This can be done in a direct way for some simple distributions or using the so-called Fast Fourier Transform (FFT) approach [7].

2.1 An Improved Algorithm

The key idea behind our approach is to rewrite the separable objective function of problem (2.3). We introduce the function $F_i : \mathbb{Z} \to R$ given by

$$F_i(n) := \mathbb{E}(\min\{n, \mathbf{D}_i\}) \tag{2.4}$$

and observe for given $n \in \mathbb{Z}_+$ that

$$F_i(n) = \sum_{j=1}^{n} \mathbb{P}\{\mathbf{D}_i \geq j\}.$$

Using this, it is obvious that F_i is a discrete concave function; i.e., the difference $F_i(n) - F_i(n-1)$ is non-increasing in n. By relation (2.4), problem (2.3) can be rewritten as

$$\begin{array}{rll} v_2(C) = & \max & \sum_{i=1}^{m} r_i F_i(x_i) \\ & \text{s.t.} & \sum_{i=1}^{m} x_i \leq C, \\ & & x \in \mathbb{Z}_+^m. \end{array}$$

Clearly, $x_i \leq C$ in this problem. Introduce now for $1 \leq j \leq C$, the values

$$\alpha_{ij} := F_i(j) - F_i(j-1) = \sum_{k=j}^{\infty} p_{ik},$$

where $p_{ik} = \mathbb{P}\{\mathbf{D}_i = k\}$. Notice that the objective function is separable. Therefore, $r_i \alpha_{ij}$ gives the *marginal* value of increasing x_i from $j-1$ to j. After this observation, we can solve problem (2.3) very fast. To explain the algorithm, we first introduce the following $m \times C$ matrix

$$\begin{bmatrix} r_1\alpha_{11} & r_1\alpha_{12} & \cdots & r_1\alpha_{1C} \\ r_2\alpha_{21} & r_2\alpha_{22} & \cdots & r_2\alpha_{2C} \\ \vdots & \vdots & \cdots & \vdots \\ r_m\alpha_{m1} & r_m\alpha_{m2} & \cdots & r_m\alpha_{mC} \end{bmatrix}. \tag{2.5}$$

Then, the optimal objective function value $v_2(C)$ can be found by sorting the $r_i\alpha_{ij}$ values, and adding up the first C terms. Consequently, the number of times index i appears among these C terms gives the optimal solution x_i^*. Notice that since F_i is discrete concave, the marginal values in each row i are in descending order; i.e., $r_i\alpha_{i1} \geq r_i\alpha_{i2} \geq \cdots \geq r_i\alpha_{iC}$. Therefore, $v_2(C)$ can be evaluated by taking the maximum of m elements C times. The computational complexity of the proposed approach reduces to the order of $O(mC)$.

2.2 A Robust Optimization Approach

To evaluate the objective function of problem (2.3), we need to know the probability distribution of the customer demand. These probabilities are usually estimated by analyzing the historical data, and hence, they are prone to inaccuracies. A reasonable consideration would be: How can we immunize the model from the inaccurate data? To answer this question, we propose next a robust modeling approach.

We assume that random variable $\mathbf{D}_i$, representing the total demand for fare class i, is concentrated on $\{0, \cdots, K\}$, and this demand has an estimated probability vector $\widehat{p}_i^\intercal = (\widehat{p}_{i0}, \cdots, \widehat{p}_{iK})$. To compensate for possible estimation errors, we consider for $1 \leq i \leq m$ the probability vectors p_i belonging to the uncertainty set P_i given by

$$P_i = \{p_i \in \mathbb{R}^{K+1} : p_i \in \widehat{p}_i + \Delta_i,\ p_i^\intercal e = 1\},$$

where

$$\Delta_i = \left\{ d_i = (d_{i0}, \cdots, d_{iK})^\intercal \in \mathbb{R}^{K+1} \mid \sum_{k=0}^{K} \left(\frac{d_{ik}}{\widehat{p}_{ik}} \right)^2 \leq \delta_i^2 \right\}$$

with $\delta_i \in [0,1]$. It is easy to verify by the positivity of $\widehat{p}_{ik}$ and the definition of Δ_i that $\widehat{p}_i + \Delta_i \subseteq \mathbb{R}_+^{K+1}$. The total demand then depends on its probability distribution p_i, and hence we denote this random variable by $\mathbf{D}_i(p_i)$. Thus, the robust counterpart of problem (2.3) is given by

$$\begin{array}{rll} v_3(C) := & \max & \sum_{i=1}^m r_i \min_{p_i \in P_i} \{\mathbb{E}\left(\min\{x_i, \mathbf{D}_i(p_i)\}\right)\} \\ & \text{s.t.} & \sum_{i=1}^m x_i \leq C, \\ & & x \in \mathbb{Z}_+^m. \end{array} \tag{2.6}$$

We introduce then the function $G_i : \mathbb{Z}_+ \to \mathbb{R}$ given by

$$G_i(n) := \min_{p_i \in P_i} \{\mathbb{E}\left(\min\{n, \mathbf{D}_i(p_i)\}\right)\}. \tag{2.7}$$

Notice for every $p_i \in P_i$ that the function

$$n \to \mathbb{E}\left(\min\{n, \mathbf{D}_i(p_i)\right)$$

is discrete concave on $\mathbb{Z}_+$. Since the point wise infimum of a collection of concave functions is again concave, the function G_i is also discrete concave on $\mathbb{Z}_+$. Then problem (2.6) can be rewritten as

$$\begin{array}{rll} v_3(C) = & \max & \sum_{i=1}^m r_i G_i(x_i) \\ & \text{s.t.} & \sum_{i=1}^m x_i \leq C, \\ & & x \in \mathbb{Z}_+^m. \end{array}$$

Observe for given $p_i \in P_i$ that

$$\mathbb{E}\left(\min\{x_i, \mathbf{D}_i(p_i)\}\right) = \sum_{k=0}^{x_i-1} k p_{ik} + x_i \sum_{k=x_i}^{K} p_{ik} = c(x_i)^\intercal p_i,$$

where

$$c(x_i)^\intercal := (c_0(x_i), c_1(x_i), \cdots, c_K(x_i)) = (0, 1, \cdots, x_i - 1, x_i, x_i, \cdots, x_i).$$

Hence, by relation (2.7), we have

$$G_i(x_i) = \min\left\{c(x_i)^\intercal p_i \mid p_i \in P_i\right\} = c(x_i)^\intercal \widehat{p}_i + \min\left\{c(x_i)^\intercal d_i \mid d_i \in \Delta_i,\ d_i^\intercal e = 0\right\}. \tag{2.8}$$

Using standard nonlinear programming techniques [2], it can be easily shown that

$$\min\{c^\intercal y \mid y^\intercal Q y \le \delta^2,\ e^\intercal y = 0\} = -\delta\sqrt{c^\intercal Q^{-1} c - \frac{(e^\intercal Q^{-1} c)^2}{e^\intercal Q^{-1} e}}, \tag{2.9}$$

where Q is symmetric and positive definite. This shows that the last term in relation (2.8) has an analytic expression. Therefore, using $c_o(x_i) = 0$ we have

$$G_i(x_i) = c(x_i)^\intercal \widehat{p}_i - \delta_i \sqrt{\sum_{k=1}^{K} \widehat{p}_{ik}^2 c_k^2(x_i) - \frac{\left(\sum_{k=1}^{K} \widehat{p}_{ik}^2 c_k(x_i)\right)^2}{\sum_{k=0}^{K} \widehat{p}_{ik}^2}}. \tag{2.10}$$

It is clear that $x_i \le C$ in problem (2.6). Introduce now for $1 \le j \le C$, the values

$$\beta_{ij} := G_i(j) - G_i(j-1).$$

Similar to the discussion in Section 2.1, we first introduce the following $m \times C$ matrix

$$\begin{bmatrix} r_1\beta_{11} & r_1\beta_{12} & \cdots & r_1\beta_{1C} \\ r_2\beta_{21} & r_2\beta_{22} & \cdots & r_2\beta_{2C} \\ \vdots & \vdots & \cdots & \vdots \\ r_m\beta_{m1} & r_m\beta_{m2} & \cdots & r_m\beta_{mC} \end{bmatrix}. \tag{2.11}$$

Then, since G_i is discrete concave, the marginal values in each row i are in descending order; i.e., $r_i\beta_{i1} \ge r_i\beta_{i2} \ge \cdots \ge r_i\beta_{iC}$. Therefore, the optimal objective function value $v_3(C)$ can be evaluated by taking the maximum of m elements C times. The computational complexity of the approach to solve (2.6) is of the order $O(mC)$.

3 Dynamic Models

Before discussing a robust version of the dynamic single-leg problem we first review the classical dynamic single-leg problem as proposed by Lautenbacher and Stidham [8]. Suppose that there are m different fare classes with the prices

$$0 < r_1 < r_2 < \cdots < r_m.$$

The no-sales class is simply represented by 0 with $r_0 = 0$. The total number of available seats is denoted by z, and the ticket sales period is partitioned into periods $1, 2, \cdots, T$. We assume that in each period either no customer is observed or at most one fare class i customer arrives. If ξ_t denotes this random demand in period t, we may assume that ξ_t may take $m+1$ different values $r_0, r_1, ..., r_m$ and its discrete density is given by

$$\mathbb{P}\{\xi_t = r_i\} = p_{it}$$

with $i = 0, 1, ..., m$ and $t = 1, ..., T$. It is also assumed that the random variables $\xi_t, t = 1, ..., T$ are independent. Introducing now the optimal random revenue $\mathbf{R}_t(z)$ that is generated from period t to T, before a request shows up in period t, while the number of available seats at the beginning of period t is z we denote by $J_t(z) := \mathbb{E}(\mathbf{R}_t(z))$ the associated expected optimal value function. Clearly $J_t(z) = \mathbb{E}_{\xi_t}(\mathbb{E}(\mathbf{R}_t(z)|\xi_t))$ and by the principle of dynamic programming it follows that

$$\mathbb{E}(\mathbf{R}_t(z)|\xi_t) = \max\{\xi_t + J_{t+1}(z-1), J_{t+1}(z)\}.$$

The above equation also yields an optimal policy: Accept the request if

$$\xi_t \geq J_{t+1}(z) - J_{t+1}(z-1).$$

Therefore,

$$J_t(z) = \mathbb{E}\left(\max\{\xi_t + J_{t+1}(z-1), J_{t+1}(z)\}\right),$$

with

$$J_T(z) = \begin{cases} \mathbb{E}(\xi_T), & \text{if } z > 0 \\ 0, & \text{if } z = 0. \end{cases}$$

For the above optimal value function, the following result has been shown [8].

Theorem 3.1 *For every given t, the function*

$$\Delta_{t+1}(z) := J_{t+1}(z) - J_{t+1}(z-1)$$

is nonnegative and non-increasing in z.

To compute the values $J_t(z)$ knowing the values $J_{t+1}(z)$ we observe

$$J_t(z) = J_{t+1}(z) + \mathbb{E}\left(\max\{\xi_t - \Delta_{t+1}(z), 0\}\right).$$

If we denote $(x)_+ = \max\{x, 0\}$, then we have

$$\mathbb{E}\left(\max\{\xi_t - \Delta_{t+1}(z), 0\}\right) = \sum_{i=0}^{m} p_{it}(r_i - \Delta_{t+1}(z))_+.$$

This yields due to $\Delta_{t+1}(z) \geq 0$ and $r_0 = 0$ that

$$J_t(z) = J_{t+1}(z) + \sum_{i=1}^{m} p_{it}(r_i - \Delta_{t+1}(z))_+. \tag{3.1}$$

A backward recursive solving requires an overall computational complexity of the order $O(mTC)$, where C is the total number of seats available. It is possible to improve this computational complexity if a careful study of the data structures is conducted.

3.1 A Robust Optimization Approach

In this case, the uncertain data in question are the estimated probability vectors $\widehat{p}_t^{\intercal} = (\widehat{p}_{1t}, \cdots, \widehat{p}_{mT})$, $t = 1, \cdots, T$. We consider the probability vectors p_t belonging to the uncertainty set P_t given by

$$P_t = \{p_t \in \mathbb{R}^m : p_t \in \widehat{p}_t + \Delta_t,\ p_t^{\intercal} e = 1\},$$

where

$$\Delta_t = \left\{ d_t = (d_{1t}, \cdots, d_{mt})^{\intercal} \in \mathbb{R}^m \mid \sum_{i=1}^{m} \left(\frac{d_{it}}{\widehat{p}_{it}}\right)^2 \leq \delta_t^2 \right\}$$

with $\delta_t \in [0, 1]$. The dynamic programming formulation then becomes

$$J_t(z) = J_{t+1}(z) + \sum_{i=1}^{m} \widehat{p}_{it}(r_i - (J_{t+1}(z) - J_{t+1}(z-1))_+ + H_t(z)$$

with

$$H_t(z) = \min \left\{ \sum_{i=1}^{m} d_{it}(r_i - (J_{t+1}(z) - J_{t+1}(z-1)))_+ : d_t \in \Delta_t, e^{\top} d_t = 0 \right\}.$$

To simplify the notation, let

$$c_{it} := (r_i - (J_{t+1}(z) - J_{t+1}(z-1)))_+, \ \ i = 1, \cdots, m.$$

Then by using relation (2.9), we have

$$H_t(z) = -\delta_t \sqrt{\sum_{i=1}^{m} \widehat{p}_{it}^2 c_{it}^2 - \frac{\left(\sum_{i=1}^{m} \widehat{p}_{it}^2 c_{it}\right)^2}{\sum_{i=1}^{m} \widehat{p}_{it}^2}}.$$

Therefore, the robust counterpart of the dynamic programming formulation becomes

$$J_t(z) = J_{t+1}(z) + \sum_{i=1}^{m} \widehat{p}_{it} c_{it} - \delta_t \sqrt{\sum_{i=1}^{m} \widehat{p}_{it}^2 c_{it}^2 - \frac{\left(\sum_{i=1}^{m} \widehat{p}_{it}^2 c_{it}\right)^2}{\sum_{i=1}^{m} \widehat{p}_{it}^2}}, \tag{3.2}$$

where $1 \leq t \leq T$ and $0 \leq z \leq C$. Since the last term in (3.2) has an analytic solution, the computational complexity of the robust approach remains the same with $O(mTC)$.

3.2 Batch Arrivals

To introduce the general case we assume in the classical dynamic leg problem that there is only one arrival at most during each time interval. That assumption may be considered restrictive. To account for multi-entry during a given time interval, let us introduce a random vector $\eta_t \in \mathbb{Z}_+^m$, where η_{it} denotes the number of customers arriving during the time interval $[t, t+1)$, $t = 1, ..., T-1$. Hence, by the dynamic programming principle we have

$$\begin{aligned} &\mathbb{E}(\mathbf{R}_t(z) \mid \eta_t = (x_1, \cdots, x_m)^{\intercal}) \\ &= \max\left(\textstyle\sum_{i=1}^{m} y_i r_i + J_{t+1}(z - \sum_{i=1}^{m} y_i) \mid 0 \leq y_i \leq x_i, i = 1, ..., m, \sum_{i=1}^{m} y_i \leq z, y \in \mathbb{Z}_+^m\right). \end{aligned}$$

Let us denote the value on the right hand side of the above equation be $R(x, z; J_{t+1})$. That is

$$\begin{array}{rcll} R(x,z;J_{t+1}) & := & \max & \sum_{i=1}^m y_i r_i + J_{t+1}(z - \sum_{i=1}^m y_i) \\ & & \text{s.t.} & \sum_{i=1}^m y_i \le z \\ & & & 0 \le y_i \le x_i,\ i = 1,...,m, \\ & & & y \in \mathbb{Z}_+^m. \end{array}$$

Using Theorem 3.1 it is easy to compute the value of $R(x, z; J_{t+1})$ for each given $z \in \mathbb{Z}_+$ and $x \in \mathbb{Z}_+^m$ with $e^T x \le z$. Compute $g(j) := J_{t+1}(z - j + 1) - J_{t+1}(z - j)$ for $j = 1, \cdots, e^T x$. Clearly, we obtain by Theorem 3.1 that $r_k - g(p) > r_k - g(q)$ for $q > p$. Notice also that $r_k - g(j) > r_l - g(j)$ for $k > l$. Therefore, the optimal objective function value can be obtained as follows: Let $s_k = \sum_{i=k}^m x_i$. Find $k = m, m-1, \cdots, 1$ such that $r_{k+1} - g(s_{k+1}) \ge 0$ and $r_k - g(s_k) < 0$. Then, find $l = 1, \cdots, x_k$ such that $r_k - g(s_k - l) \ge 0$ and $r_k - g(s_k - l + 1) < 0$. The optimal solution then becomes $y_i = x_i$ for $i = k+1, \cdots, m$, $y_k = l$, and $y_i = 0$ for $i = 1, \cdots, k-1$. This yields the optimal objective function value

$$R(x, z; J_{t+1}) = \sum_{i=k+1}^{m} r_i x_i + l r_k - \sum_{j=1}^{s_k - l} g(j).$$

The above procedure is illustrated in Figure 1 and summarized in Algorithm 3.1. Notice that the marginal gain decreases as y_k, $1 \le k \le m$ increases and the procedure starts with the most profitable fare class m.

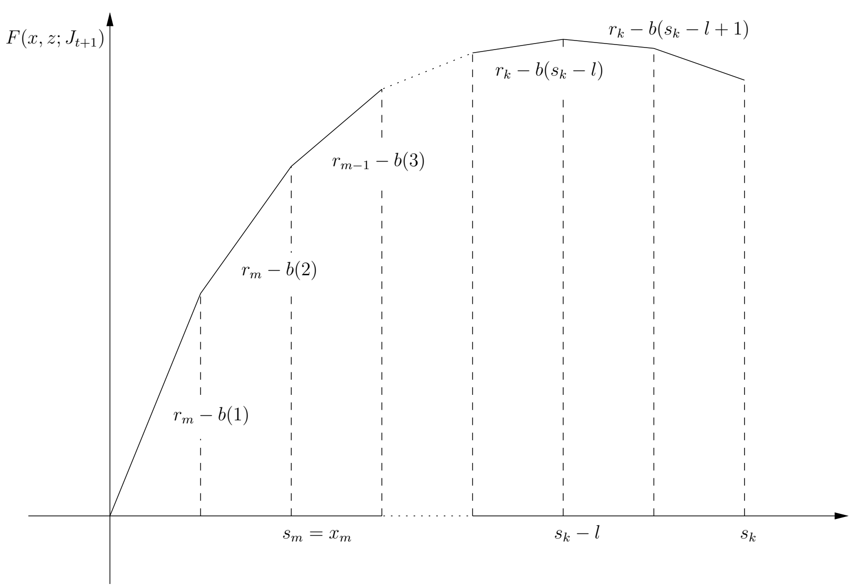

Figure 1: The calculation of $R(x, z; J_{t+1})$.

The dynamic programming recursion is

$$J_t(z) = \mathbb{E}_\eta \left(\mathbf{R}(\eta, z; J_{t+1}) \right), \tag{3.3}$$

where $t = 1, 2, ..., T$, and $z = 0, 1, ..., C$. In case the number of the fare classes, m, is relatively small, then a straightforward computation yields a solution at the complexity bound of $O(mTC^{m+1})$.

Algorithm 3.1 The algorithm for calculating $R(x, z; J_{t+1})$

1. **Initialize:** $y_i = 0$, $1 \le i \le m$, and $k = m$.
2. Set $s_k = \sum_{i=k}^{m} x_i$.
3. If $r_k - g(s_k) \ge 0$ then set $y_k = x_k$, $k = k - 1$ and go to Step 2; otherwise, set $l = 0$.
4. While $r_k - g(s_k - l) < 0$ set $l = l + 1$ and $y_k = l$.
5. **Output:**
$$R(x, z; J_{t+1}) = \sum_{i=k+1}^{m} r_i x_i + l r_k - \sum_{j=1}^{s_k - l} g(j).$$

Clearly, $R(x, z; J_{t+1})$ is *monotonic* in x for fixed z and $t+1$; i.e., if $x', x'' \in \mathbb{Z}_+^m$ satisfying $x' \le x''$, then $R(x', z; J_{t+1}) \le R(x'', z; J_{t+1})$. It also has the following *lexicographic* property: if $x', x'' \in \mathbb{Z}_+^m$ with $e^T x' = e^T x'' \le z$ differ only in two components, say, $x'_k > x''_k$ and $x'_l < x''_l$ with $l > k$, then it holds that $R(x', z; J_{t+1}) \le R(x'', z; J_{t+1})$.

To reduce the computational complexity, we may consider for instance a two-point distribution for each fare class customers. That is, we let l_{it} and u_{it} be respectively the minimum and the maximum amount of arriving customers for the fare class i during the time interval t. The dynamic programming then requires a computational complexity of $O(mTC2^m)$. In the case of airline revenue management, typically $m \le 16$, and so for a flight with 400 seats and decision period $T = 12$, the computation complexity is in the order of 10^9 basic operations: a large but manageable number. If m falls in a reasonable range, say $m = 5$, then we may afford to consider a finer grid of scenarios, say we may consider a 10-point distribution for each fare class without losing tractability.

4 The Solution with Perfect Information

A useful concept in decision analysis is *perfect information*. Although this type of information rarely exists, it provides an upper bound on the value of real information since it pictures the "best case" scenario [4]. In our static problem setting, perfect information implies elimination of uncertainty about the total demand for each fare class. The subsequent model focuses on the perfect information from this "a priori" perspective. In Section 5, we solve the perfect information model approximately and compare our results with the results that we obtain after solving the other models of the previous sections.

Suppose that we decide on the allocation after knowing all the realized demands. Then, we obtain the following optimization model

$$v_4(C) := \mathbb{E}\left(\max\left\{\sum_{i=1}^{m} r_i \min\{x_i, \mathbf{D}_i\} \mid \sum_{i=1}^{m} x_i \le C, x_i \in \mathbb{Z}_+\right\}\right). \tag{4.1}$$

It is obvious that $v_2(C) \le v_4(C)$. We may consider the positive difference $v_4(C) - v_2(C)$ as the expected cost of having *incomplete information*. We now introduce both the partial sum $\mathbf{S}_n :=$

$\sum_{j=n}^{m} \mathbf{D}_j$ with $\mathbf{D}_0 := 0$ and the stochastic process $\mathbf{N}(C) := \min\{0 \le n \le m \mid \mathbf{S}_n \le C\}$. Then by relation (2.2), the random optimal solution $(\mathbf{x}_i^*)_{i=1}^n$ for the random demands $\mathbf{D}_i$, $1 \le i \le m$ is given by

$$\mathbf{x}_i^* = \begin{cases} \mathbf{D}_i, & \text{if } i \ge \mathbf{N}(C) \\ C - \mathbf{S}_{\mathbf{N}(C)}, & \text{if } i = \mathbf{N}(C) - 1 \\ 0, & \text{if } i < \mathbf{N}(C) - 1. \end{cases}$$

The associated random optimal objective value equals

$$\mathbf{v}_1(C) = \sum_{i=\mathbf{N}(C)}^{m} r_i \mathbf{D}_i + \left(C - \mathbf{S}_{\mathbf{N}(C)}\right) r_{\mathbf{N}(C)-1}.$$

As in the deterministic case, for each realization this is a concave function in C. This shows that

$$v_4(C) = \mathbb{E}\left(\sum_{i=\mathbf{N}(C)}^{m} r_i \mathbf{D}_i + \left(C - \mathbf{S}_{\mathbf{N}(C)}\right) r_{\mathbf{N}(C)-1}\right). \tag{4.2}$$

In general, it seems to be difficult to give an analytical expression for this expectation and so we might approximate the above expectation by means of the Monte Carlo method [14].

5 Simulation Experiments

To support our theoretical study, we conduct simulation experiments and report our observations in this section. We first compare, in the first subsection, the non-robust static model (2.3) with its robust counterpart (2.6). In the second subsection, a similar study is carried out to compare the non-robust dynamic model (3.1) with its counterpart (3.2). To see the differences between the static and the dynamic modeling approaches, we conduct additional simulation experiments in the final subsection. Using the same data, we also approximate the expectation in the perfect information model (4.1). We give then the comparison among static, dynamic and perfect information models. In all our simulation experiments we have used MATLAB 7.0 on a personal computer with 1.5 GHz Intel Celeron M processor and 256 MB of RAM.

5.1 Static Models: Non-robust vs. Robust

We have implemented the algorithm given in Section 2.1. Recall that the same algorithm can also be applied to solve the robust version given in Section 2.2. As shown in relation (2.10) the convex subproblem has an analytic solution. Therefore, the only difference between the non-robust and robust implementations is the calculation of the $m \times C$ matrices given by (2.5) and (2.11), respectively.

We take M simulation runs with different seeds. In each simulation run, we first generate the estimated probability vectors $\widehat{p}_i \in \mathbb{R}^{K+1}$, $1 \le i \le m$. Then we use the algorithm discussed in Section 2.1 to find the optimal seat allocations of different fare classes for both the non-robust and the robust models. We next generate N realizations of the probability vectors $p_i \in \mathbb{R}^{K+1}$ uniformly from P_i, $1 \le i \le m$. Notice that to find these p_i vectors, one needs to generate uniform samples from the intersection of an ellipsoid and a hyperplane. This issue is discussed in Appendix A. After generating the probability vectors p_i, $1 \leq i \leq m$, by Algorithm A.1, we simulate the demand for each fare class according to these probabilities. The total revenues are then evaluated according to the non-robust and the robust seat allocations. As our statistics, we store the mean and the standard deviation of the N realized revenues.

We assume that $\delta_i = 1$, $1 \leq i \leq m$. This reflects the "conservative" choice of the decision maker, where the estimation errors can be large. The distribution of the demand for each fare class i, $1 \leq i \leq m$ is assumed to be a truncated Poisson distribution with parameters $\lambda_i > 0$ and K. Consequently, the total demand for fare class i is concentrated on $\{0, \cdots, K\}$. Moreover, the distribution parameters are sorted in descending order $\lambda_1 > \lambda_2 > \cdots > \lambda_m$ to reflect the higher demand for relatively cheaper fare class seats. In each run, the parameters λ_i are uniformly generated from the intervals $[\kappa_i, \mu_i]$, $1 \leq i \leq m$. The actual values of the parameters that we use in our simulation are given in Table 1.

Table 1: The parameters used in the simulation of static models.

Parameters	**Values**
$[M, N, K, C, m]$	[25, 250, 100, 100, 4]
(r_1, r_2, r_3, r_4)	(2, 3, 4, 6)
$(\kappa_1, \kappa_2, \kappa_3, \kappa_4)$	(40, 20 ,10 ,1)
$(\mu_1, \mu_2, \mu_3, \mu_4)$	(70, 40 ,30 ,10)

Table 2 shows the simulation results. The first column of the table gives the run numbers. The averages over N realized revenues for non-robust and robust models are reported in columns two and three, respectively. The fourth column gives the relative differences between the non-robust and robust revenues in percentages. Similarly, the standard deviations over N realized revenues for non-robust and robust models are reported in columns five and six, respectively. The last column shows the relative differences in percentages. The runs 7, 8 and 24 do not show any difference between the corresponding non-robust and robust models because for both models the optimal allocations turned out to be the same. It is clear from the fourth column of Table 2 that the non-robust model yields slightly better revenue than the robust version. However, as shown in the last column the solution found by the robust model has, in most cases, significantly less deviation than the non-robust version. Therefore, we find a stable solution at the expense of a small decrease in the revenue. The small difference in the total revenues does not come as a surprise, since it can be easily shown that the conservative solution found by the robust approach yields a revenue less than the value found by solving the non-robust model.

Since the convex subproblem has an analytic solution, the computational time between solving the robust and the non-robust models is insignificant. Moreover, the simulation with the above parameters (for 25 runs) takes on average less than 3 minutes. Therefore, we do not report our computation times separately. This remark is valid for all the subsequent results that we report.

Table 2: The simulation results for the static models.

	Mean			**Standard Deviation**		
Run	**Robust**$^{(a)}$	**Non-robust**$^{(b)}$	**%** $100(b-a)/b$	**Robust**$^{(c)}$	**Non-robust**$^{(d)}$	**%** $100(d-c)/d$
1	278.3200	278.4000	0.0287	17.8700	19.0900	6.3939
2	292.5200	292.6300	0.0383	15.8350	18.2530	13.2480
3	263.8100	263.8900	0.0318	11.9040	13.5140	11.9130
4	289.4200	290.1200	0.2413	16.6830	19.6300	15.0120
5	268.8000	269.0000	0.0744	13.1890	15.5750	15.3140
6	286.9000	287.5200	0.2170	17.9980	19.5710	8.0349
7	260.3300	260.3300	0.0000	20.5020	20.5020	0.0000
8	234.9700	234.9700	0.0000	22.5590	22.5590	0.0000
9	286.8700	287.5800	0.2462	15.6510	18.2360	14.1760
10	273.5900	273.8400	0.0906	14.3170	15.5240	7.7740
11	275.3300	275.6800	0.1291	15.6000	16.5110	5.5180
12	285.4900	286.1900	0.2432	12.9710	15.8860	18.3550
13	259.9800	260.0200	0.0154	18.0090	18.7750	4.0777
14	275.6000	276.5600	0.3500	12.8040	14.9030	14.0810
15	277.1200	277.7400	0.2218	11.9320	14.0740	15.2160
16	287.7000	288.2400	0.1887	13.1010	15.2410	14.0350
17	283.8500	284.5100	0.2334	13.1380	15.3610	14.4730
18	299.5600	299.7600	0.0681	17.5140	17.6740	0.9024
19	304.3500	305.3700	0.3340	16.9290	19.6080	13.6630
20	285.9600	286.3300	0.1313	13.2190	15.7330	15.9770
21	289.0400	289.6900	0.2237	15.5990	18.5220	15.7780
22	268.0600	268.1600	0.0403	15.2080	15.5560	2.2364
23	291.6600	292.1000	0.1506	14.8070	17.3390	14.6020
24	264.7100	264.7100	0.0000	23.4670	23.4670	0.0000
25	261.2900	261.4400	0.0581	15.1350	15.4880	2.2761

5.2 Dynamic Models: Non-robust vs. Robust

We have implemented a dynamic programming algorithm to solve (3.1). Since the convex subproblem of the robust model (3.2) has an analytic solution, only the calculation of the return at each stage is changed, and hence, the dynamic programming algorithm implemented for the non-robust model (3.1) is slightly modified to solve the robust version (3.2).

As in the previous subsection, we take M simulation runs with different seeds. In each simulation run, we first generate the estimated probability vectors $\widehat{p}_t \in \mathbb{R}^m$, $1 \leq t \leq T$. Then we compute the non-robust and the robust optimal policies by the corresponding dynamic programming algorithms. Using Algorithm A in Appendix A, we generate N realizations of the probability vectors $p_t \in \mathbb{R}^m$ uniformly from P_t, $1 \leq t \leq T$. Given a realization p_t, we simulate S times the arrival process, and then, using the non-robust and robust optimal policies, we compute the corresponding optimal seat allocations. As our statistics, we store the mean and the standard deviation of the $N \times S$ realized revenues.

Again, we take $\delta_t = 1$ for all $1 \leq t \leq T$. The probability vector $\widehat{p}_t$ of period t is assumed to be Dirichlet distributed with parameters γ_{it}, $0 \leq i \leq m$. This distribution allows us to generate realizations that add up to 1, and therefore, we have valid arrival probabilities at each period t for the fare classes. Notice also that, the parameters of the Dirichlet distribution change at every period t. We assume, as the departure time T approaches, that the requests for cheaper fare classes reduce, whereas the requests for the more expensive fare classes increase. The details of this implementation are given in Appendix B. The actual values of the parameters that we use in our simulation are given in Table 3.

Table 3: The parameters used in the simulation of dynamic models.

Parameters	**Values**
$[M, N, S, C, T, m]$	[25, 25, 10, 100, 200, 4]
(r_1, r_2, r_3, r_4)	(2, 3, 4, 6)
$[\bar{v}_0, \bar{v}, v_0, v_1, v_2, v_3, v_4]^*$	[1, 2, 3, 5, 4 ,1, 0.5]

* See Appendix B for details.

Similar to previous subsection, we report our results in Table 4. The columns have the same meaning as in Table 2. The figures, however, are reported over $N \times S$ realized revenues. Our results with the dynamic model intensify our observations with the static model. Again the non-robust model yields slightly better revenues than the robust version. Nevertheless, as shown in the last column the solution found by the non-robust model yields a substantial deviation.

Table 4: The simulation results for the dynamic models.

	Mean			**Standard Deviation**		
Run	**Robust**$^{(a)}$	**Non-robust**$^{(b)}$	**%** $100(b-a)/b$	**Robust**$^{(c)}$	**Non-robust**$^{(d)}$	**%** $100(d-c)/d$
1	432.6600	437.3400	1.0692	13.0110	13.7500	5.3766
2	438.1000	443.0200	1.1088	11.8790	15.3450	22.5850
3	425.0600	427.3000	0.5252	12.8420	14.9320	13.9940
4	437.3300	444.0200	1.5071	11.8860	13.7100	13.3050
5	430.9800	435.9200	1.1314	12.3960	14.5080	14.5550
6	427.4600	432.5900	1.1854	11.5500	14.9910	22.9550
7	425.1600	430.3700	1.2092	12.7460	15.4330	17.4100
8	429.7400	436.3800	1.5198	12.0690	14.7410	18.1240
9	424.4900	428.8000	1.0047	12.2520	13.7000	10.5710
10	436.9900	441.6200	1.0480	12.4890	15.5960	19.9190
11	432.2000	438.5200	1.4412	13.1890	14.9990	12.0700
12	439.3000	445.0900	1.3013	12.3520	15.3690	19.6310
13	429.7000	432.5800	0.6658	12.5240	15.2760	18.0190
14	422.6300	425.8800	0.7627	12.4780	13.3760	6.7065
15	435.7100	439.6200	0.8899	11.8290	15.1270	21.8030
16	433.0700	439.2400	1.4061	12.0510	14.2880	15.6600
17	435.6600	439.6900	0.9170	13.4070	14.6110	8.2456
18	426.5100	431.5600	1.1688	11.9030	13.9600	14.7340
19	432.2600	436.7400	1.0240	11.3040	14.2460	20.6540
20	428.3600	431.8700	0.8114	12.7750	13.3550	4.3444
21	426.9800	431.8800	1.1364	11.8540	15.1900	21.9600
22	439.4600	444.4500	1.1232	12.9070	15.4250	16.3230
23	426.9400	430.7400	0.8804	12.9640	14.9580	13.3320
24	432.3400	437.7300	1.2309	12.3330	14.5600	15.2960
25	429.2500	436.4500	1.6488	12.6650	14.1260	10.3490

5.3 Cost of Incomplete Information

In this subsection we conduct simulation experiments to compare the static model (2.3), the dynamic model (3.1) and the perfect information model (4.1). The main motivation of these experiments is to check the effect of having additional information as one has more information in the dynamic model than the static model, and similarly, as the perfect information model includes more information than the dynamic model.

We take M simulation runs with different seeds. In each simulation run, we first generate for $1 \leq t \leq T$ the arrival probability vector $p_t \in \mathbb{R}^m_+$ from a Dirichlet distribution with parameters γ_{it}, $0 \leq i \leq m$. As we discussed in Section 4, it is difficult to compute $v_4(C)$ and solve (4.1) to optimality. Therefore, we implemented a Monte Carlo algorithm, which generates N demand realizations according to p_t, $1 \leq t \leq T$, and then gives a point estimate of (4.2). Next, we compute the expected optimal revenue by the dynamic model (3.1). To make a fair comparison between the static and the other two models, we need to compute the demand probabilities $p_{ik} = \mathbb{P}\{\mathbf{D}_i = k\}$, $1 \leq k \leq T$, by using the arrival probabilities p_t, $1 \leq t \leq T$. Since $p_{it} = \mathbb{P}\{\xi_t = r_i\}$, $0 \leq i \leq m$, $1 \leq t \leq T$, we have

$$\mathbf{D}_i = \sum_{t=1}^{T} \mathbf{1}_{\{\xi_t = r_i\}}.$$

Since it is assumed that the random variables ξ_t, $1 \leq t \leq T$, are independent it follows that the Bernoulli random variables $1_{\{\xi_t = r_i\}}$, $1 \leq t \leq T$, are also independent. This shows for every $\alpha \in (0, 2\pi)$ that the discrete Fourier transform $\mathcal{P}(\alpha) = \mathbb{E}(\exp(\mathrm{i}\alpha\mathbf{D}_i))$ has the form

$$\mathcal{P}(\alpha) = \mathbb{E}(\,\exp(\mathrm{i}\alpha(\textstyle\sum_{t=1}^{T} \mathbf{1}_{\{\xi_t = r_i\}}))) = \Pi_{t=1}^{T}\mathbb{E}(\,\exp(\mathrm{i}\alpha\mathbf{1}_{\{\xi_t = r_i\}})).$$

Consequently,

$$\mathbb{E}(\exp(\mathrm{i}\alpha\mathbf{1}_{\{\xi_t = r_i\}})) = p_{it}\exp(\mathrm{i}\alpha) + (1 - p_{it}) = 1 - p_{it}(1 - \exp(\mathrm{i}\alpha))$$

and so, we obtain

$$\mathcal{P}(\alpha) = \Pi_{t=1}^{T}(1 - p_{it}(1 - \exp(\mathrm{i}\alpha)).$$

It is well known that

$$p_{ik} = \frac{1}{T+1}\sum_{n=0}^{T}\mathcal{P}(\frac{2\pi n}{T+1})\exp(\frac{-2\pi \mathrm{i} nk}{T+1}).$$

By using the FFT algorithm of the order $O(T\log T)$, one can easily recover the probabilities p_{ik} [7]. After recovering these probabilities, we compute the expected optimal revenue by the static model (2.3). As our statistics, we store the estimated total revenue of the perfect information model and the expected optimal revenues found by dynamic and static models, respectively. The parameters we use in our experiments are the same as in Table 3 except the parameter S is not required and $N = 1000$.

Table 5 shows the simulation results. The second column gives a point estimate of the optimal value of the perfect information model over N trials. The third and fourth columns include the revenues found by the dynamic and static models, respectively. The fifth column shows the relative differences between the perfect information model and the dynamic model in percentages. Similarly, the last column gives the relative differences between the perfect information model and the static model. As expected, the model with the perfect information yields higher revenues than both the dynamic and the static models. However, as the fifth column shows, the cost of incomplete information is rather insignificant when the dynamic model is considered. On the other hand, the cost of incomplete information increases as one prefers the static model over the dynamic version.

Table 5: The simulation results for the perfect information, static and dynamic models.

Run	**Perfect**$^{(a)}$	**Dynamic**$^{(b)}$	**Static**$^{(c)}$	$100(a-b)/a$	$100(a-c)/a$
1	429.0100	427.0300	410.7100	0.4622	4.2666
2	434.2700	432.5000	416.0400	0.4068	4.1983
3	432.2200	430.4100	413.6400	0.4179	4.2990
4	436.5800	434.9000	417.8100	0.3852	4.3001
5	438.1600	436.1400	419.4700	0.4612	4.2660
6	443.5300	441.5500	424.5700	0.4484	4.2762
7	431.6700	430.5000	413.7800	0.2701	4.1437
8	435.7300	434.6000	417.3700	0.2607	4.2145
9	433.0000	431.0500	414.3100	0.4495	4.3152
10	439.1600	437.5400	420.3800	0.3689	4.2776
11	439.1100	437.3000	420.3500	0.4122	4.2723
12	433.9600	432.8600	416.0800	0.2528	4.1208
13	433.0600	431.9100	415.2600	0.2665	4.1104
14	437.8800	437.2700	420.4100	0.1376	3.9897
15	435.7800	435.4300	418.6100	0.0807	3.9400
16	438.0900	436.6900	419.8000	0.3203	4.1761
17	433.6800	432.1600	415.5000	0.3502	4.1921
18	442.6800	440.4200	423.2100	0.5096	4.3969
19	436.7900	435.0000	418.2700	0.4095	4.2407
20	440.7800	439.2500	422.2800	0.3482	4.1968
21	433.7500	431.7600	415.2800	0.4592	4.2579
22	439.8400	438.1700	421.1800	0.3796	4.2412
23	432.9500	431.7200	415.4000	0.2855	4.0547
24	433.6200	432.4200	415.1600	0.2766	4.2556
25	439.3300	436.7800	420.0900	0.5801	4.3810

6 Conclusion

In this study we have shown by means of simulation that the use of robust versions of the classical static and dynamic single leg seat allocation problems in airline revenue management may be worthwhile due to the reduction in variability of the generated revenues. This reduction is much larger as the reduction in average revenue due to the conservative behavior of the considered robust models. In a subsequent paper we will consider extensions of the models in the network environment.

APPENDIX

A Uniform Sampling from The Uncertainty Set

Notice that in both static and dynamic model simulation runs, we need to generate sample vectors p_i, $1 \leq i \leq m$ and p_t, $1 \leq t \leq T$, from the intersection of an ellipsoid and a hyperplane of appropriate dimensions. In our subsequent discussion, we omit for ease of notation the subindices i and t.

To conduct our simulation experiments, we need to generate sample vectors p from the set

$$P = \{p \in \mathbb{R}^q_+ \mid p \in \widehat{p} + \Delta, p^\mathsf{T} e = 1\},$$

where

$$\Delta = \left\{ x \in \mathbb{R}^q \mid \sum_{j=1}^{q} \left(\frac{x_j}{\widehat{p}_j} \right)^2 \leq \delta^2 \right\}.$$

Notice that $\widehat{p}^\mathsf{T} e = 1$. Therefore, if we generate uniform samples from the set

$$S = \left\{ x \in \mathbb{R}^q \mid \sum_{j=1}^{q} \left(\frac{x_j}{\widehat{p}_j} \right)^2 \leq \delta^2,\ x^\mathsf{T} e = 0 \right\},$$

then we can set $p = \widehat{p} + x$. Notice that S defines an ellipsoid on a $q-1$ dimensional subspace (see Figure 2). It is not straightforward to generate uniform samples from S. However, it is shown by Fang. et. al. that uniform samples can be easily generated from unit hyper-spheres [6, Section 3.1.5]. Therefore, we next apply two transformations so that we can transform S to a $q-1$ dimensional unit hypersphere.

Let $y = Ax$, where A is a $q \times q$ diagonal matrix with nonzero elements $(1/(\delta\widehat{p}_1)), \cdots, 1/(\delta\widehat{p}_q))$. Using this transformation, the set S becomes

$$S_y = \{y \in \mathbb{R}^q \mid y^\mathsf{T} y \leq 1,\ y^\mathsf{T} \widehat{p} = 0\}.$$

Since we want to focus only on the unit hypersphere, we further apply a transformation to reflect the vector $u := (\widehat{p}/\|\widehat{p}\|) - I_1$, where I_1 is the unit vector corresponding to the first column of the identity matrix I. This transformation is called Householder reflection [7], and it is applied by using the orthonormal matrix

$$B = I - \frac{2}{u^\mathsf{T} u} u u^\mathsf{T}.$$

Using now $z = By$, the set S_y becomes

$$S_z = \{z \in \mathbb{R}^q \mid z^\mathsf{T} z \leq 1,\ z_1 = 0\}.$$

Notice that it is now enough to generate a realization of the vector $\mathbf{Z} = (\mathbf{Z}_1, \mathbf{Z}_2, \cdots, \mathbf{Z}_q)$ uniformly from S_z. Then, using $B^{-1} = B^\mathsf{T}$ and the Jacobian transformation theorem, $\mathbf{X} = A^{-1} B^{-1} Z = A^{-1} B^\mathsf{T} \mathbf{Z}$ yields a uniformly distributed vector from S as desired.

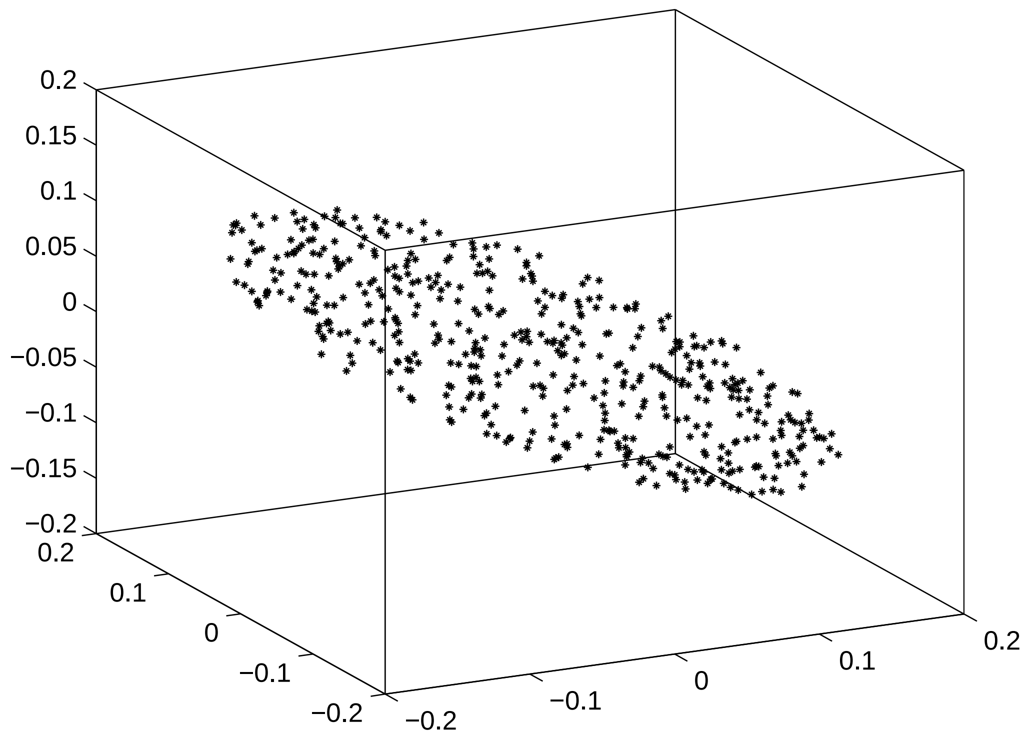

Figure 2: A set of uniform samples from the ellipsoid centered at $\widehat{p}^{\mathsf{T}} = (0.5, 0.2, 0.3)$ with $\delta = 1$.

To generate a realization of the vector $\mathbf{Z}$ from S_z, observe that we can equivalently generate a realization of the vector $\bar{\mathbf{Z}} = (\mathbf{Z}_2, \cdots, \mathbf{Z}_q)$ uniformly from the $q-1$ dimensional unit hypersphere

$$\bar{S}_z = \{z = (z_2, \cdots, z_q) \in \mathbb{R}^{q-1} \mid z^{\mathsf{T}} z \leq 1\}.$$

It is given on page 75 of [6] that the random vector $\bar{\mathbf{Z}} = \mathbf{R}\mathbf{Q}$ is uniformly distributed on $\bar{S}_z$, where $\mathbf{Q}$ is a $q-1$ dimensional random vector distributed on the boundary of $\bar{S}_z$, $\mathbf{R}$ is a random variable with the distribution function

$$\mathbb{P}\{\mathbf{R} \leq r\} = r^{q-1},\ 0 \leq r \leq 1,$$

and the random variables $\mathbf{R}$ and $\mathbf{Q}$ are independent. Clearly by the inverse transformation method we obtain that $\mathbf{R} =^d \mathbf{U}^{(q-1)^{-1}}$ with $\mathbf{U}$ uniform distributed on $(0, 1)$. To generate a realization of the random vector $\mathbf{Q} = (\mathbf{Q}_1, \cdots, \mathbf{Q}_{q-1})$, we can generate for the components $\mathbf{Q}_i$, $1 \leq i \leq q-1$, independent standard normal variates and then normalize the resulting vector [6]. The following algorithm summarizes the steps to generate uniform samples from the set S. An illustrative set of samples generated by this algorithm is given in Figure 2.

Algorithm A.1 Generating uniform samples from S

1. Generate standard normal variates $N_1, \cdots, N_{q-1}$ and a random number U.

2. Let $N = (N_1, N_2, \cdots, N_{q-1})$ and set

$$\bar{z} = \left(\frac{U^{(q-1)^{-1}} N_1}{\|N\|}, ..., \frac{U^{(q-1)^{-1}} N_{q-1}}{\|N\|} \right)$$

3. Set $z := \begin{pmatrix} 0 \\ \bar{z} \end{pmatrix}$ and return $x = A^{-1} B^{\intercal} z$.

B Generating Arrival Probabilities for The Dynamic Models

In our simulation of the dynamic models, we generate the probability vectors $\widehat{p}_t^{\intercal} = (\widehat{p}_{0t}, \widehat{p}_{1t}, ..., \widehat{p}_{mt})$, $1 \leq t \leq T$ in the following way:

1. Generate some numbers $v_i, 0 \leq i \leq m$ and $\bar{v}_0, \bar{v}$ satisfying $0 < \bar{v}_0 < \bar{v} < v_0$, $0 < v_m < v_{m-1} < ... < v_1$ and $v_m < \bar{v} < v_1$.

2. Introduce the functions $\alpha_i : \mathbb{R}_+ \to \mathbb{R}$, $0 \leq i \leq m$ given by

$$\gamma_i(t) = v_i + (\bar{v} - v_i)(1 - \exp(-\frac{mt}{T})), \;\; 1 \leq i \leq m$$

   and

$$\gamma_0(t) = v_0 + (\bar{v}_0 - v_0)(1 - \exp(-\frac{mt}{T})).$$

3. Introduce the random vector $\mathbf{X} = (\mathbf{X}_1, ..., \mathbf{X}_T) \in \mathbb{R}_+^{(m+1)\times T}$ consisting of the random vectors

$$\mathbf{X}_t = (\mathbf{X}_{0t}, ..., \mathbf{X}_{mt}), \; 1 \leq t \leq T$$

   with the random variable $\mathbf{X}_{it}$, $0 \leq i \leq m$, $1 \leq t \leq T$ independent, and for each (i, t), the random variable $\mathbf{X}_{it}$ has a gamma distribution with scale parameter 1 and shape parameter $\gamma_i(t)$.

4. Introduce now for each (i, t)

$$\widehat{p}_{it} = \frac{\mathbf{X}_{it}}{\sum_{j=0}^{m} \mathbf{X}_{jt}}.$$

   It can be shown that the above procedure generates realizations $\widehat{p}_t$ of a Dirichlet distributed random vector $\widehat{p}_t$ with parameters $\gamma_0(t), \cdots, \gamma_m(t)$ [6]. This yields that

$$\mathbb{E}(\widehat{p}_{it}) = \frac{\gamma_i(t)}{\sum_{j=0}^{m} \gamma_j(t)}.$$

   Introducing now $i^* = \min\{1 \leq i \leq m \mid v_i > \bar{v}\}$ it follows by the definition of the function γ_i that the function γ_i is increasing for $i > i^*$ and decreasing for $i < i^*$. This modeling approach tries the capture the practical assumption that the arrival intensities are decreasing for the cheaper fare classes $i < i^*$ in the total remaining time before departure of the plane (but always above the arrival intensities of the more expensive fare classes $i \geq i^*$), while for the more expensive fare classes $i \geq i^*$ are increasing in the remaining time before departure. Figure 3 illustrates the change of the distribution parameters over time. Observe for t large enough and $1 \leq i \leq m$ that

$$\mathbb{E}(\widehat{p}_{it}) \approx \frac{\bar{v}_i}{\sum_{j=0}^{m} \bar{v}_j}$$

and $t \mapsto \mathbb{E}(\widehat{p}_{it})$ is increasing in t for $i > i^*$ and decreasing for $i \leq i^*$.

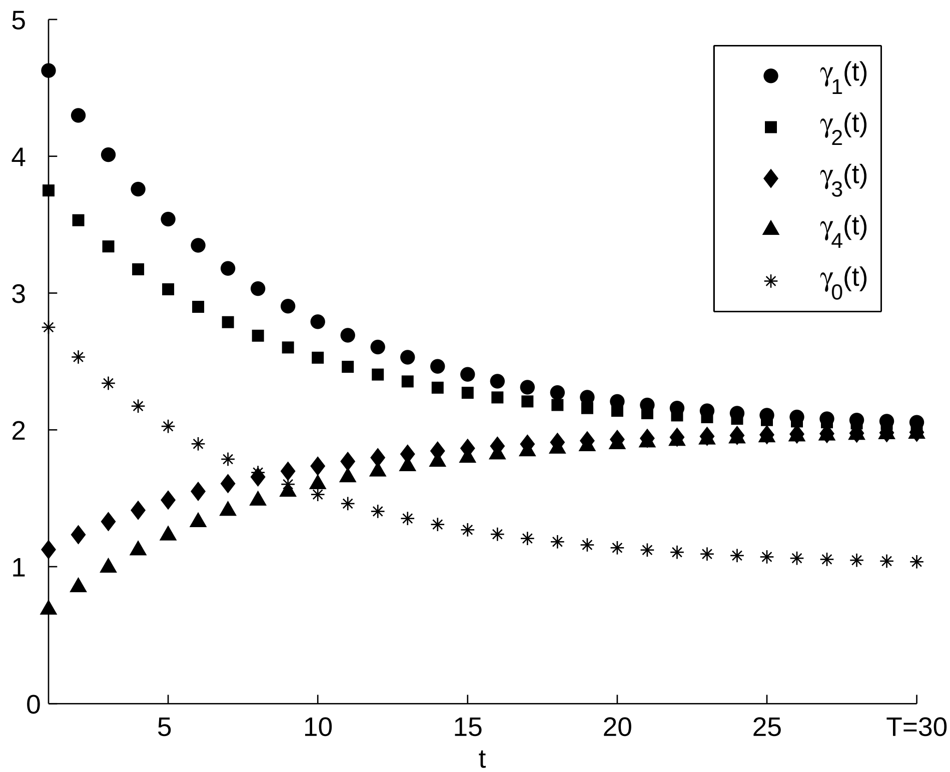

Figure 3: The change of distribution parameters over time ($i^* = 3$, $m = 4$, $T = 30$ and $[\bar{v}_0, \bar{v}, v_0, v_1, v_2, v_3, v_4]^* = [1, 2, 3, 5, 4, 1, 0.5]$).

References

[1] Belobaba, P.P. Applications of a probabilistic decision model to airline seat inventory control. *Operations Research*, 37:183–197, 1989.

[2] Bertsekas, D.P. *Nonlinear Programming (2nd Edition)*. Athena Scientific, Belmont, 1999.

[3] Brumelle, S.L and J.I. McGill. Airline seat allocation with multiple nested fare classes. *Operations Research*, 41:127–137, 1993.

[4] Clemen R.T. and T. Reilly. *Making Hard Decisions with Decision Tools, 2nd Edition*. Duxbury, 2003.

[5] De Boer, S.V., Freling, R. and N. Piersma. Mathematical programming for network revenue management revisited. *European Journal of Operational Research*, 137(1):72–92, 2002.

[6] Fang, K.-T., Kotz, S. and K.W. Ng. *Symmetric Multivariate and Related Distributions*. Chapman and Hall, New York, 1990.

[7] Golub, G.H. and C.F. van Loan. *Matrix Computations, 3rd Edition*. The Johns Hopkins University Press, Baltimore, 1996.

[8] Lautenbacher, C.J. and S. Stidham, jr. The underlying markov decision process in the single-leg airline yield-management problem. *Transportation Science*, 33(2):136–146, 1999.

[9] Lee,T.C and Hersch, M. A model for dynamic airline seat inventory control with multiple seat bookings. *Transportation Science*, 27:252–265, 1993.

[10] Littlewood, K. Forecasting and control of passengers. *Proceedings of the 12th Annual AGIFORS Symposium*, pages 95–117, 1972.

[11] McGill, J.I. and G.J. van Ryzin. Revenue management: research overview and prospects. *Transportation Science*, 33(2):233–256, 1999.

[12] Richter, H. The differential revenue method to determine optimal seat allotments by fare type. *Proceedings of the 22nd Annual AGIFORS Symposium*, pages 339–362, 1982.

[13] Robinson, L.W. Optimal and approximate control policies for airline booking with sequential nonmonotonic fare classes. *Operations Research*, 43:252–263, 1995.

[14] Ross, S.M. *Simulation (3rd Edition)*. Academic Press, New York, 2002.

[15] Talluri, K.M. and G.J. van Ryzin. *The Theory and Practice of Revenue Management*. Kluwer Academic Publishers, Dordrecht, 2004.

[16] Walczak, D. and S. Brumelle. Dynamic airline revenue management with multiple semi-markov demand. *Operations Research*, 51:137–148, 2003.

[17] Wollmer, R.D. A hub-spoke seat management model. Technical report, Mc Donell Douglas Co., Long Beach, CA, 1986.

[18] Wollmer, R.D. An airline seat management problem for a single leg when lower fare classes book first. *Operations Research*, 40:26–37, 1992.