A Dynamic Programming Decomposition Method for Making Overbooking Decisions over an Airline Network

Alexander Erdelyi
School of Operations Research and Information Engineering,
Cornell University, Ithaca, New York 14853, USA
alex@orie.cornell.edu

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

June 1, 2009

Abstract

In this paper, we develop a revenue management model to jointly make the capacity allocation and overbooking decisions over an airline network. Our approach begins with the dynamic programming formulation of the capacity allocation and overbooking problem and uses an approximation strategy to decompose the dynamic programming formulation by the flight legs. This decomposition idea opens up the possibility of obtaining approximate solutions by concentrating on one flight leg at a time, but the capacity allocation and overbooking problem that takes place over a single flight leg still turns out to be intractable. We use a state aggregation approach to obtain high quality solutions to the single leg problem. Overall, our model constructs separable approximations to the value functions, which can be used to make the capacity allocation and overbooking decisions for the whole airline network. Computational experiments indicate that our model performs significantly better than a variety of benchmark strategies from the literature.

Capacity allocation and overbooking are two main ingredients of network revenue management. In particular, capacity allocation deals with the question of which itineraries to keep open for purchase and which itineraries to close as the remaining capacities on the flight legs are depleted over time with the customer purchases. Overbooking deals with the question of to what extent the sales should exceed the physically available capacity on the flight legs, given that not everyone with a reservation ends up showing up at the departure time. The capacity allocation and overbooking decisions are inherently connected. What fare classes to make available for purchase depends on how many seats in excess of the physically available capacity the airline is willing to sell. On the other hand, how much to overbook depends on what itineraries the airline keeps open and the probability that a customer who purchases a reservation for one of the open itineraries shows up at the departure time.

In this paper, we propose a revenue management model that makes the joint capacity allocation and overbooking decisions over an airline network. Our approach formulates the problem as a dynamic program and uses an approximation strategy to decompose the dynamic programming formulation by the flight legs. This decomposition idea opens up the possibility of obtaining approximate solutions by concentrating on one flight leg at a time, though the capacity allocation and overbooking problem that takes place over a single flight leg still happens to be intractable. In particular, the state variable in the dynamic programming formulation of the single leg problem involves a large number of dimensions in practical applications. We overcome this difficulty by using state aggregation to obtain high quality solutions to the single leg capacity allocation and overbooking problem. Ultimately, our model provides separable approximations to the value functions, which can be used to construct a capacity allocation and overbooking policy for the whole airline network.

Our work in this paper draws on two streams of literature. The first stream of literature is the work on dynamic programming decomposition methods in network revenue management. Dynamic programming decomposition methods date back to Belobaba (1987) and they are approximate methods aimed at decomposing the network revenue management problem by the flight legs. The basic idea is to associate a displacement adjusted fare with each itinerary over each flight leg, which is different from the actual fare that the airline charges. The displacement adjusted fares immediately allow us to solve a sequence of single leg revenue management problems. In the single leg revenue management problem that takes place over a particular flight leg, we only concentrate on the remaining capacity on this flight leg and assume that the fares associated with the itineraries are equal to the displacement adjusted fares over this flight leg. Once we have solved the single leg problem over each flight leg, we add up the value functions obtained for different flight legs to obtain separable approximations to the value functions. In this paper, we use a similar idea to decompose the capacity allocation and overbooking problem. The important distinction of our paper is that we explicitly deal with overbooking, whereas the earlier decomposition methods work exclusively under the assumption that overbooking is not possible and all reservations show up at the departure time. Our extension to overbooking is nontrivial and has important practical implications as overbooking plays a major role in airline operations.

The second stream of literature that we draw on is the work on solving the capacity allocation and overbooking problem over a single flight leg. This stream of literature becomes especially useful when we try to solve the single leg capacity allocation and overbooking problems after decomposing the original problem. As mentioned above, the single leg capacity allocation and overbooking problem is intractable, as its dynamic programming formulation involves a high dimensional state variable. To be able to solve this problem, we build on the approach proposed by Subramanian et al. (1999). In particular, we assume that the proportions of the reservations that we have for different itineraries are fixed and known. This allows us to keep track of only the total number of reservations, rather than the number of reservations for each itinerary, in our dynamic programming formulation. In this case, the state variable in the dynamic programming formulation of the single leg problem collapses to a scalar and the dynamic programming formulation becomes tractable.

In this paper, we make the following research contributions. 1) We develop a model to make the capacity allocation and overbooking decisions over an airline network. The idea behind our model is to decompose the problem by the flight legs and to solve a sequence of single leg problems. 2) We show that our approach provides an upper bound on the optimal total expected profit as long as we can solve the single leg problems accurately. 3) However, noting that the capacity allocation and overbooking problem over a single flight leg is still intractable, we show how to obtain approximate solutions to the single leg problems in a tractable manner. 4) Computational experiments indicate that our model performs significantly better than many of the existing models in the literature.

The rest of the paper is organized as follows. In Section 1, we review the other literature that is related to our work. In Section 2, we give a dynamic programming formulation for the capacity allocation and overbooking problem over an airline network. In Section 3, we give a simple deterministic linear program that can be used to develop control policies. In Section 4, we describe our model by using the deterministic linear program as a starting point. In Section 5, we present our computational experiments. In Section 6, we provide concluding remarks.

1 Review of Related Literature

Despite the fact that there is substantial literature on capacity allocation over an airline network, the interaction between capacity allocation and overbooking is not thoroughly studied. Early models focus on the single leg version of the problem. Beckmann (1958), Thompson (1961) and Coughlan (1999) develop single leg capacity allocation and overbooking models under the assumption that the demands from different fare classes are static random variables. These models ignore the temporal dynamics of the demand process and their goal is to decide how many seats to allocate to different fare classes. Later models by Chatwin (1992), Chatwin (1999) and Subramanian et al. (1999) also consider the single leg problem, but they try to capture the dynamics of the demand process more accurately. Subramanian et al. (1999) note the intractability of the dynamic programming formulation of the single leg problem and propose the approximation strategy that we build on in this paper. It is important to contrast their observation with the single leg capacity allocation problem without overbooking. If overbooking is not allowed, then the dynamic programming formulation of the capacity allocation problem involves a scalar state variable and can easily be solved. Therefore, the possibility of overbooking, by itself, brings nontrivial challenges even for the single leg case. Karaesmen and van Ryzin (2004*b*) describe an overbooking model for multiple flight legs that operate between the same origin destination pair and can serve as substitutes of each other.

There are a few papers that consider the overbooking decisions over an airline network. A popular method to make the capacity allocation decisions in network revenue management problems is to solve a deterministic linear program. This linear program is built under the assumption that the itinerary requests are known in advance and they take on their expected values. There is a constraint associated with each flight leg in the linear program and the right sides of these constraints are the leg capacities. Therefore, the optimal values of the dual variables associated with these constraints are used to estimate the opportunity cost of a seat on different flights legs. In this case, one can construct a capacity control policy, where the fare from an itinerary request is compared with the total opportunity cost of the capacities that would be consumed by this itinerary request. If the fare exceeds the total opportunity cost, then the itinerary request is accepted. There are many variants of the linear programming idea and Section 3.3 in Talluri and van Ryzin (2004) describes these variants. We do not go into the details of these variants, as most of them deal only with capacity allocation decisions. However, Bertsimas and Popescu (2003) show how to build a deterministic linear program to deal with overbooking and no shows. We use a variant of their deterministic linear program as a benchmark strategy in our computational experiments. Karaesmen and van Ryzin (2004*a*) develop a capacity allocation and overbooking model, where they compute booking limits by using the optimal objective value of the deterministic linear program as an estimate of the total expected revenue from the itinerary requests. Gallego and van Ryzin (1997) provide theoretical support for the deterministic linear program by showing that the control policy obtained from a variant of the deterministic linear program is asymptotically optimal as the leg capacities and the expected number of itinerary requests increase linearly at the same rate. Kleywegt (2001) constructs a pricing and overbooking model in continuous time. The demand process that he uses is deterministic and he utilizes Lagrangian duality to solve the model.

The literature on decomposition of network revenue management problems is also related to our paper. Williamson (1992) is one of the first to decompose the network revenue management problem by the flight legs. Her goal is to apply the expected marginal seat revenue heuristic of Belobaba (1987) on each flight leg individually to construct value function approximations. Section 3.4.4 in Talluri and van Ryzin (2004) describes a more refined variant of her approach in the sense that this variant does not assume that the demand from different fare classes arrive over nonoverlapping time intervals. Liu and van Ryzin (2008) and Bront et al. (2008) show how to extend decomposition methods to address the customer choice behavior among the different itineraries that are available for purchase. Topaloglu (2006) shows that decomposition methods can be visualized as an application of Lagrangian relaxation to the dynamic programming formulation of the network revenue management problem. Kunnumkal and Topaloglu (2007) show that it is possible to extend the observations of Topaloglu (2006) to address the customer choice behavior. There is some recent work on decomposing the capacity allocation and overbooking problem over an airline network. Erdelyi and Topaloglu (2009) use separable functions to approximate the value functions in the dynamic programming formulation of the capacity allocation and overbooking problem. In this respect, their paper is connected to our work. However, their approximations are separable by the itineraries and the number of scalar functions they keep is equal to the number of possible itineraries. In contrast, our approximations are separable by the flight legs and the number of scalar functions we keep is equal to the number of flight legs. The number of flight legs is generally smaller than the number of itineraries. Furthermore, they use simulation to construct their scalar functions, whereas we solve small dynamic programs to construct our value function approximations. We use their model as a benchmark strategy.

There are recent approaches for the capacity allocation problem over an airline network. Adelman (2007) uses linear value function approximations for the capacity allocation problem and he chooses the slopes and intercepts of the value function approximations by solving a linear program that represents the dynamic programming formulation of the problem. Zhang and Adelman (2006) extend this approach to deal with the customer choice behavior. They also show that decomposition methods can provide upper bounds on the optimal total expected revenue. Meissner and Strauss (2008) refine the approach proposed by Adelman (2007) by using piecewise linear value function approximations. An important advantage of the recent approaches is that they provide upper bounds on the optimal total expected revenue. However, the recent approaches do not address the interaction between capacity allocation and overbooking and the goal of our paper is to fill this gap.

2 Problem Formulation

We consider a set of flight legs over an airline network that can be used to serve the itinerary requests arriving randomly over time. At each time period, an itinerary request arrives and we need to decide whether to accept or reject this itinerary request. An accepted itinerary request becomes a reservation, whereas a rejected itinerary request simply leaves the system. At the departure time of the flight legs, a certain portion of reservations shows up and we need to decide which of these reservations should be allowed boarding. The objective is to maximize total expected profit defined as the difference between the expected revenue obtained by accepting itinerary requests and the expected penalty cost incurred by denying boarding to reservations.

The set of flight legs is $\mathcal{L}$ and the set of itineraries is $\mathcal{J}$. We note that a flight leg is referred to as a resource and an itinerary is referred to as a product in some settings. The problem takes place over the finite planning horizon $\{\tau, \ldots, 0\}$. The itinerary requests arrive over time periods $\mathcal{T} = \{\tau, \ldots, 1\}$ and the flights depart at time period 0. The probability that there is a request for itinerary j at time period t is p_{jt}. Accepting a request for itinerary j generates a revenue of f_j and this reservation shows up at the departure time with probability q_j. If a reservation for itinerary j shows up at the departure time and it is denied boarding, then we incur deny penalty cost of θ_j. If we allow boarding to a reservation for itinerary j, then we consume a_{ij} units of capacity on flight leg i. The capacity on flight leg i is c_i. We assume that the arrivals of the itinerary requests at different time periods and the show up decisions of different reservations at the departure time are independent. We also assume that the reservations are not canceled over time periods $\{\tau, \ldots, 1\}$ and we do not give refunds to the no shows, but these assumptions are for brevity and one can make extensions to address cancellations and refunds.

We let x_{jt} denote the total number of reservations for itinerary j at the beginning of time period t so that $x_t = \{x_{jt} : \ j \in \mathcal{J}\}$ captures the state of the reservations. Assuming that the number of reservations for itinerary j at the beginning of time period 0 is x_{j0}, we use $S_j(x_{j0})$ to denote the number of reservations for itinerary j that show up at the departure time. Given the assumption that the show up decisions of different reservations are independent, $S_j(x_{j0})$ has a binomial distribution with parameters (x_{j0}, q_j). If we use $S(x_0) = \{S_j(x_{j0}) : j \in \mathcal{J}\}$ to denote the state of the reservations that show up at the departure time, then we can compute the penalty cost associated with the denied reservations by solving the problem

$$\begin{aligned} \Gamma(S(x_0)) \;=\; \min \quad & \sum_{j \in \mathcal{J}} \theta_j \, w_j && && (1) \\ \text{subject to} \quad & \sum_{j \in \mathcal{J}} a_{ij} \, [S_j(x_{j0}) - w_j] \le c_i && i \in \mathcal{L} && (2) \\ & w_j \le S_j(x_{j0}) && j \in \mathcal{J} && (3) \\ & w_j \in \mathbb{Z}_+ && j \in \mathcal{J}, && (4) \end{aligned}$$

where w_j is the number of reservations for itinerary j that we deny boarding. The objective function of the problem above corresponds to the penalty costs associated with the denied reservations. Constraints (2) ensure that the reservations that we allow boarding do not exceed the leg capacities, whereas constraints (3) ensure that the numbers of denied reservations do not exceed the numbers of reservations that show up at the departure time. It is important to observe that problem (1)-(4) assumes that we can jointly decide which reservations should be denied boarding throughout the network and this can be an optimistic assumption. Letting e_j be the $|\mathcal{J}|$ dimensional unit vector with a one in the element corresponding to j, we can find the optimal policy by computing the value functions through the optimality equation

$$V_t(x_t) = \sum_{j \in \mathcal{J}} p_{jt} \max\{f_j + V_{t-1}(x_t + e_j),\, V_{t-1}(x_t)\} + \Big[1 - \sum_{j \in \mathcal{J}} p_{jt}\Big] V_{t-1}(x_t) \tag{5}$$

with the boundary condition that $V_0(x_0) = -\mathbb{E}\{\Gamma(S(x_0))\}$. In this case, if the state of the reservations at the beginning of time period t is given by x_t, then it is optimal to accept a request for itinerary j at time period t whenever

$$f_j \ge V_{t-1}(x_t) - V_{t-1}(x_t + e_j). \tag{6}$$

Unfortunately, even for modest sized applications, the state vector x_t involves hundreds of dimensions rendering exact solution to the optimality equation in (5) computationally intractable. In next section, we begin by describing an approximate solution method that involves solving a deterministic linear program. Following this, we build on the deterministic linear program to develop a more sophisticated approximate solution method.

3 Deterministic Linear Program

A standard solution method for the network revenue management problem described in the previous section involves solving a deterministic linear program. This linear program is formulated under the assumption that the arrivals of the itinerary requests and the show up decisions of the reservations take on their expected values. In particular, if we let z_j be the number of requests for itinerary j that we plan to accept over the planning horizon and w_j be the number of reservations that we plan to deny boarding, then this linear program can be formulated as

$$\max \quad \sum_{j \in \mathcal{J}} f_j \, z_j - \sum_{j \in \mathcal{J}} \theta_j \, w_j \tag{7}$$

$$\text{subject to} \quad \sum_{j \in \mathcal{J}} a_{ij} \, [q_j \, z_j - w_j] \leq c_i \qquad i \in \mathcal{L} \tag{8}$$

$$z_j \leq \sum_{t \in \mathcal{T}} p_{jt} \qquad j \in \mathcal{J} \tag{9}$$

$$w_j - q_j \, z_j \leq 0 \qquad j \in \mathcal{J} \tag{10}$$

$$z_j, \, w_j \geq 0 \qquad j \in \mathcal{J}. \tag{11}$$

In the problem above, we assume that if we accept z_j requests for itinerary j, then $q_j \, z_j$ reservations for itinerary j show up at the departure time. Constraints (8) ensure that the numbers of reservations that we allow boarding do not exceed the leg capacities. Constraints (9) ensure that the numbers of itinerary requests that we accept do not exceed the expected numbers of itinerary requests. Constraints (10) ensure that the numbers of denied reservations do not exceed the expected numbers of reservations that show up at the departure time. The deterministic linear programming formulation for the network revenue management problem is widely known under the assumption that overbooking is not possible and all reservations show up at the departure time; see Talluri and van Ryzin (1998). Problem (7)-(11) extends this formulation to handle overbooking and no shows. Although this extension is quite intuitive, to our knowledge, Bertsimas and Popescu (2003) is the only reference to this extension.

One use of problem (7)-(11) is that its dual solution can be used to construct a policy to accept or reject the itinerary requests. Letting $\{\lambda_i^* : i \in \mathcal{L}\}$ be optimal values of the dual variables associated with constraints (8) in problem (7)-(11), the idea is to use λ_i^* to estimate the opportunity cost of a unit of capacity on flight leg i. In this case, if the revenue from an itinerary request exceeds the total expected opportunity cost of the capacities consumed by this itinerary request or if the revenue from an itinerary request exceeds the expected penalty cost, then we accept the itinerary request. In other words, if we have

$$f_j \geq \min \Big\{ q_j \sum_{i \in \mathcal{L}} a_{ij} \, \lambda_i^*, \, q_j \, \theta_j \Big\}, \tag{12}$$

then we accept a request for itinerary j. The two arguments of the $\min\{\cdot, \cdot\}$ operator above capture two effects. If the total expected opportunity cost of the capacities consumed by a request for itinerary j is small enough that we have $f_j \geq q_j \sum_{i \in \mathcal{L}} a_{ij} \, \lambda_i^*$, then we accept a request for itinerary j. Furthermore, if we have $f_j \geq q_j \, \theta_j$, then we can generate revenue, in expectation, simply by accepting a request for itinerary j and denying boarding to this reservation at the departure time. We refer to the decision rule in (12) as the DLP policy, standing for deterministic linear program. This decision rule is also used by Bertsimas and Popescu (2003).

One other use of problem (7)-(11) is that its optimal objective value provides an upper bound on the optimal total expected profit. In other words, letting z_{LP} be the optimal objective value of problem (7)-(11) and $\bar{0}$ be the $|\mathcal{J}|$ dimensional vector of zeros, it is possible to show that $V_\tau(\bar{0}) \leq z_{LP}$. For future reference, we state this result as a proposition below. The proof of this proposition can be found in Erdelyi and Topaloglu (2009).

Proposition 1 *We have* $V_\tau(\bar{0}) \leq z_{LP}$.

The upper bound in Proposition 1 can be useful when assessing the optimality gap of a suboptimal decision rule such as the DLP policy in (12).

4 Dynamic Programming Decomposition

There are several shortcomings of the deterministic linear program. It only uses the total expected numbers of the itinerary requests, ignoring the probability distributions and the temporal dynamics of the arrivals of the itinerary requests. Furthermore, it assumes that the numbers of reservations that show up at the departure time take on their expected values. In this section, we build on the deterministic linear program to develop a solution method that captures the temporal dynamics of the itinerary requests somewhat more accurately.

4.1 Decomposing into Single Leg Revenue Management Problems

The starting point for our approach is a duality argument on the deterministic linear program to decompose the network revenue management problem into a sequence of single leg revenue management problems. We begin letting $\{\lambda_i^* : i \in \mathcal{L}\}$ be the optimal values of the dual variables associated with constraints (8) in problem (7)-(11). We choose an arbitrary flight leg i and relax constraints (8) in problem (7)-(11) for all other flight legs by associating the dual multipliers $\{\lambda_l^* : l \in \mathcal{L} \setminus \{i\}\}$. In this case, linear programming duality implies that problem (7)-(11) has the same optimal objective value as the problem

$$\begin{aligned} \max \quad & \sum_{j \in \mathcal{J}} \Big[f_j - q_j \sum_{l \in \mathcal{L} \setminus \{i\}} a_{lj}\, \lambda_l^* \Big] z_j - \sum_{j \in \mathcal{J}} \Big[\theta_j - \sum_{l \in \mathcal{L} \setminus \{i\}} a_{lj}\, \lambda_l^* \Big] w_j + \sum_{l \in \mathcal{L} \setminus \{i\}} \lambda_l^*\, c_l \\ \text{subject to} \quad & \sum_{j \in \mathcal{J}} a_{ij}\, [q_j\, z_j - w_j] \leq c_i \\ & (9), (10), (11). \end{aligned}$$

We note that the problem above includes the capacity constraint only for flight leg i. For notational brevity, we let

$$\Lambda_j^i = \sum_{l \in \mathcal{L} \setminus \{i\}} a_{lj}\, \lambda_l^* \qquad F_j^i = f_j - q_j\, \Lambda_j^i \qquad \Theta_j^i = \theta_j - \Lambda_j^i. \tag{13}$$

Omitting the constant term $\sum_{l \in \mathcal{L} \setminus \{i\}} \lambda_l^*\, c_l$, we write the problem above as

$$\max \quad \sum_{j \in \mathcal{J}} F_j^i\, z_j - \sum_{j \in \mathcal{J}} \Theta_j^i\, w_j \tag{14}$$

$$\text{subject to} \quad \sum_{j \in \mathcal{J}} a_{ij}\, [q_j\, z_j - w_j] \leq c_i \tag{15}$$

$$(9), (10), (11), \tag{16}$$

in which case, the optimal objective value of problem (14)-(16) differs from z_{LP} by $\sum_{l \in \mathcal{L} \setminus \{i\}} \lambda_l^* \, c_l$.

The decision variables z_j and w_j do not appear in constraint (15) whenever itinerary j does not use the capacity on flight leg i. This observation allows us to decompose problem (14)-(16) into two problems, one of which involves the itineraries that use the capacity on flight leg i and the other one involves the remaining itineraries. To this end, we let $\mathcal{J}^i = \{j \in \mathcal{J} : a_{ij} > 0\}$ so that $\mathcal{J}^i$ is the set of itineraries that use the capacity on flight leg i. In this case, it is easy to see that the optimal objective value of problem (14)-(16) is equal to the sum of the optimal objective values of the problem

$$\begin{aligned}
\max \quad & \sum_{j \in \mathcal{J}^i} F_j^i \, z_j - \sum_{j \in \mathcal{J}^i} \Theta_j^i \, w_j && (17)\\
\text{subject to} \quad & \sum_{j \in \mathcal{J}^i} a_{ij} \, [q_j \, z_j - w_j] \leq c_i && (18)\\
& z_j \leq \sum_{t \in \mathcal{T}} p_{jt} \qquad j \in \mathcal{J}^i && (19)\\
& w_j - q_j \, z_j \leq 0 \qquad j \in \mathcal{J}^i && (20)\\
& z_j, \, w_j \geq 0 \qquad j \in \mathcal{J}^i, && (21)
\end{aligned}$$

which involves only the decision variables $\{z_j : j \in \mathcal{J}^i\}$ and $\{w_j : j \in \mathcal{J}^i\}$, and the problem

$$\begin{aligned}
\max \quad & \sum_{j \in \mathcal{J} \setminus \mathcal{J}^i} F_j^i \, z_j - \sum_{j \in \mathcal{J} \setminus \mathcal{J}^i} \Theta_j^i \, w_j && (22)\\
\text{subject to} \quad & z_j \leq \sum_{t \in \mathcal{T}} p_{jt} \qquad j \in \mathcal{J} \setminus \mathcal{J}^i && (23)\\
& w_j - q_j \, z_j \leq 0 \qquad j \in \mathcal{J} \setminus \mathcal{J}^i && (24)\\
& z_j, \, w_j \geq 0 \qquad j \in \mathcal{J} \setminus \mathcal{J}^i, && (25)
\end{aligned}$$

which involves only the decision variables $\{z_j : j \in \mathcal{J} \setminus \mathcal{J}^i\}$ and $\{w_j : j \in \mathcal{J} \setminus \mathcal{J}^i\}$. It turns out that the optimal objective value of problem (22)-(25) can easily be obtained by mere inspection. In particular, we show in Appendix A that the optimal objective value of problem (22)-(25) is equal to $\sum_{t \in \mathcal{T}} \sum_{j \in \mathcal{J} \setminus \mathcal{J}^i} p_{jt} \, \max\{F_j^i, F_j^i - q_j \, \Theta_j^i, 0\}$. Therefore, summing up the discussion so far in this section, if we let z_{LP}^i be the optimal objective value of problem (17)-(21), then we have

$$z_{LP} = z_{LP}^i + \sum_{t \in \mathcal{T}} \sum_{j \in \mathcal{J} \setminus \mathcal{J}^i} p_{jt} \, \max\{F_j^i, F_j^i - q_j \, \Theta_j^i, 0\} + \sum_{l \in \mathcal{L} \setminus \{i\}} \lambda_l^* \, c_l. \tag{26}$$

Comparing problem (17)-(21) with problem (7)-(11), we can observe that problem (17)-(21) is the deterministic linear program corresponding to a single leg revenue management problem that takes place over flight leg i. In this single leg revenue management problem, only the requests for the itineraries in the set $\mathcal{J}^i$ are considered. If we accept a request for itinerary j, then we generate a revenue of F_j^i. If we deny boarding to a reservation for itinerary j, then we incur a penalty cost of Θ_j^i. Noting that the optimal objective value of problem (17)-(21) is denoted by z_{LP}^i, Proposition 1 implies that z_{LP}^i provides an upper bound on the optimal total expected profit for the single leg revenue management problem that takes place over flight leg i.

On the other hand, we can compute the optimal total expected profit for the above described single leg revenue management problem taking place over flight leg i by solving the corresponding dynamic program. To this end, we introduce some new notation. We let $\mathcal{R}^i(\cdot)$ be the operator that restricts the components of a $|\mathcal{J}|$ dimensional vector to those that correspond to the elements of $\mathcal{J}^i$. For example, we have $\mathcal{R}^i(x_t) = \{x_{jt} : j \in \mathcal{J}^i\}$ and $\mathcal{R}^i(S(x_0)) = \{S_j(x_{j0}) : j \in \mathcal{J}^i\}$. In this case, the optimality equation for the single leg revenue management problem that takes place over flight leg i reads

$$V_t^i(\mathcal{R}^i(x_t)) = \sum_{j \in \mathcal{J}^i} p_{jt} \max\{F_j^i + V_{t-1}^i(\mathcal{R}^i(x_t + e_j)), V_{t-1}^i(\mathcal{R}^i(x_t))\} + \Big[1 - \sum_{j \in \mathcal{J}^i} p_{jt}\Big] V_{t-1}^i(\mathcal{R}^i(x_t)) \quad (27)$$

with the boundary condition that $V_0^i(\mathcal{R}^i(x_0)) = -\mathbb{E}\{\Gamma^i(\mathcal{R}^i(S(x_0)))\}$. Here, $\Gamma^i(\cdot)$ accounts for the penalty cost of denied boarding at the departure time in the single leg revenue management problem that takes place over flight leg i and it is given by

$$\Gamma^i(\mathcal{R}^i(S(x_0))) = \min \sum_{j \in \mathcal{J}^i} \Theta_j^i \, w_j \quad (28)$$

$$\text{subject to} \quad \sum_{j \in \mathcal{J}^i} a_{ij} \, [S_j(x_{j0}) - w_j] \le c_i \quad (29)$$

$$w_j \le S_j(x_{j0}) \qquad j \in \mathcal{J}^i \quad (30)$$

$$w_j \in \mathbb{Z}_+ \qquad j \in \mathcal{J}^i. \quad (31)$$

We recall that z_{LP}^i provides an upper bound on the optimal total expected profit for the single leg revenue management problem that takes place over flight leg i. This optimal total expected profit is given by $V_\tau^i(\mathcal{R}^i(\bar{0}))$ so that we obtain $V_\tau^i(\mathcal{R}^i(\bar{0})) \le z_{LP}^i$. The next proposition shows the relationship between the solutions to the optimality equations in (5) and (27). Its proof is in Appendix B.

Proposition 2 *For all $t \in \mathcal{T}$, we have*

$$V_t(x_t) \le V_t^i(\mathcal{R}^i(x_t)) - \sum_{j \in \mathcal{J}^i} q_j \, \Lambda_j^i \, x_{jt} - \sum_{j \in \mathcal{J} \setminus \mathcal{J}^i} \min\Big\{q_j \sum_{l \in \mathcal{L}} a_{lj} \, \lambda_l^*, \, q_j \, \theta_j\Big\} x_{jt}$$
$$+ \sum_{j \in \mathcal{J} \setminus \mathcal{J}^i} \sum_{s=1}^{t} p_{js} \, \max\{F_j^i, F_j^i - q_j \, \Theta_j^i, 0\} + \sum_{l \in \mathcal{L} \setminus \{i\}} \lambda_l^* \, c_l. \quad (32)$$

Using Proposition 2 with $t = \tau$ and $x_t = \bar{0}$, the discussion just before this proposition implies that

$$V_\tau(\bar{0}) \le V_\tau^i(\mathcal{R}^i(\bar{0})) + \sum_{j \in \mathcal{J} \setminus \mathcal{J}^i} \sum_{s=1}^{\tau} p_{js} \, \max\{F_j^i, F_j^i - q_j \, \Theta_j^i, 0\} + \sum_{l \in \mathcal{L} \setminus \{i\}} \lambda_l^* \, c_l$$
$$\le z_{LP}^i + \sum_{j \in \mathcal{J} \setminus \mathcal{J}^i} \sum_{s=1}^{\tau} p_{js} \, \max\{F_j^i, F_j^i - q_j \, \Theta_j^i, 0\} + \sum_{l \in \mathcal{L} \setminus \{i\}} \lambda_l^* \, c_l = z_{LP},$$

where the last equality follows from (26). Therefore, we can obtain an upper bound on the optimal total expected profit by solving the optimality equation in (27) and this upper bound is tighter than the one provided by the optimal objective value of problem (7)-(11). Nevertheless, the state variable in the optimality equation in (27) has still $|\mathcal{J}^i|$ dimensions, which can be quite large for many practical applications. Before we describe one method to approximate the solution to this optimality equation, we take a quick detour in the next section and describe how we can use the upper bound in Proposition 2 to construct a policy to accept or reject the itinerary requests.

4.2 Approximating the Optimal Decision Rule

Proposition 2 suggests approximating $V_t(x_t)$ with the upper bound given by the expression on the right side of (32). In particular, using $\tilde{V}_t^i(x_t)$ to denote the expression on the right side of (32), we can replace $V_{t-1}(x_t) - V_{t-1}(x_t + e_j)$ in the decision rule in (6) with $\tilde{V}_{t-1}^i(x_t) - \tilde{V}_{t-1}^i(x_t + e_j)$ and follow this decision rule to accept or reject the itinerary requests. One ambiguous aspect of this approach is that the choice of flight leg i is arbitrary and the performance of the proposed decision rule can depend on the choice of this flight leg. We work around this ambiguity by computing $\{\tilde{V}_t^i(\cdot) : t \in \mathcal{T}\}$ for all $i \in \mathcal{L}$ so that we can use the average $\sum_{i\in\mathcal{L}} \tilde{V}_t^i(x_t)/|\mathcal{L}|$ as an approximation to $V_t(x_t)$. Noting that $\tilde{V}_t^i(x_t) \geq V_t(x_t)$ for all $i \in \mathcal{L}$, we still have the upper bound that $\sum_{i\in\mathcal{L}} \tilde{V}_t^i(x_t)/|\mathcal{L}| \geq V_t(x_t)$. Thus, we propose approximating $V_{t-1}(x_t) - V_{t-1}(x_t + e_j)$ on the right side of (6) by $\sum_{i\in\mathcal{L}} \tilde{V}_{t-1}^i(x_t)/|\mathcal{L}| - \sum_{i\in\mathcal{L}} \tilde{V}_{t-1}^i(x_t + e_j)/|\mathcal{L}|$. The definition of $\tilde{V}_t^i(x_t)$ in (32) implies that

$$\tilde{V}_{t-1}^i(x_t) - \tilde{V}_{t-1}^i(x_t + e_j) = \begin{cases} V_{t-1}^i(\mathcal{R}^i(x_t)) - V_{t-1}^i(\mathcal{R}^i(x_t + e_j)) + q_j\,\Lambda_j^i & \text{if } j \in \mathcal{J}^i \\ \min\Big\{q_j \sum_{l\in\mathcal{L}} a_{lj}\,\lambda_l^*,\; q_j\,\theta_j\Big\} & \text{if } j \in \mathcal{J} \setminus \mathcal{J}^i. \end{cases}$$

Therefore, letting $\mathbf{1}(\cdot)$ be the indicator function, if the state of the reservations at time period t is given by x_t, then we accept a request for itinerary j whenever we have

$$f_j \geq \frac{1}{|\mathcal{L}|}\sum_{i\in\mathcal{L}} \mathbf{1}(j \in \mathcal{J}^i)\Big\{V_{t-1}^i(\mathcal{R}^i(x_t)) - V_{t-1}^i(\mathcal{R}^i(x_t + e_j)) + q_j\,\Lambda_j^i\Big\} + \frac{1}{|\mathcal{L}|}\sum_{i\in\mathcal{L}} \mathbf{1}(j \in \mathcal{J}\setminus\mathcal{J}^i)\,\min\Big\{q_j\sum_{l\in\mathcal{L}} a_{lj}\,\lambda_l^*,\; q_j\,\theta_j\Big\}. \tag{33}$$

One possible way to look at the decision rule in (33) is that each flight leg contributes one term to the expression on the right side. If flight leg i is used by itinerary j, then this flight leg contributes the term $V_{t-1}^i(\mathcal{R}^i(x_t)) - V_{t-1}^i(\mathcal{R}^i(x_t + e_j)) + q_j\,\Lambda_j^i$. If, on the other hand, flight leg i is not used by itinerary j, then this flight leg contributes the term $\min\{q_j \sum_{l\in\mathcal{L}} a_{lj}\,\lambda_l^*,\, q_j\,\theta_j\}$. The important observation is that the term $\min\{q_j \sum_{l\in\mathcal{L}} a_{lj}\,\lambda_l^*,\, q_j\,\theta_j\}$ is identical to the right side of the DLP policy in (12). Therefore, the flight legs that are not used by itinerary j do not provide any additional information over what is already provided by the deterministic linear program. Furthermore, the number of flight legs that are not used by itinerary j is likely to be substantially larger than the number of flight legs that are used by itinerary j, which implies that the right side of the expression above is likely to be dominated by the term $\min\{q_j \sum_{l\in\mathcal{L}} a_{lj}\,\lambda_l^*,\, q_j\,\theta_j\}$. Thus, one conjectures that the decision rule in (33) performs very much like the DLP policy. A small set of computational experiments confirmed this conjecture.

To overcome this shortcoming, instead of averaging over all flight legs and using $\sum_{i\in\mathcal{L}} \tilde{V}_t^i(x_t)/|\mathcal{L}|$ as an approximation to $V_t(x_t)$, we average only over the flight legs that are used by a particular itinerary.

In particular, we let $\mathcal{L}^j = \{i \in \mathcal{L} : a_{ij} > 0\}$ so that $\mathcal{L}^j$ is the set of flight legs that are used by itinerary j. In this case, whenever we need to make a decision for itinerary j, we use $\sum_{i \in \mathcal{L}^j} \tilde{V}_t^i(x_t)/|\mathcal{L}^j|$ as an approximation to $V_t(x_t)$. We note that we still have the upper bound that $\sum_{i \in \mathcal{L}^j} \tilde{V}_t^i(x_t)/|\mathcal{L}^j| \geq V_t(x_t)$. Thus, if the state of the reservations at time period t is given by x_t, then we accept a request for itinerary j whenever we have

$$f_j \geq \frac{1}{|\mathcal{L}^j|} \sum_{i \in \mathcal{L}^j} \Big\{ V_{t-1}^i(\mathcal{R}^i(x_t)) - V_{t-1}^i(\mathcal{R}^i(x_t + e_j)) + q_j \, \Lambda_j^i \Big\}. \tag{34}$$

The state variable in the optimality equation in (27) has $|\mathcal{J}^i|$ dimensions. Theoretically, this is an improvement in comparison to the optimality equation in (5), which involves a state variable with $|\mathcal{J}|$ dimensions. Practically, however, this improvement is irrelevant as $|\mathcal{J}^i|$ is on the order of hundreds or thousands even for modest applications. Therefore, it is still quite difficult to compute the value functions $\{V_t^i(\cdot) : t \in \mathcal{T}\}$ and to use the decision rule in (34). In the next section, we give one method to approximate the value functions $\{V_t^i(\cdot) : t \in \mathcal{T}\}$, which seems to work particularly well for our application context.

4.3 Reducing the State Space

In this section, we consider the single leg revenue management problem that takes place over flight leg i whose dynamic programming formulation is given in (27). Our goal is to approximate the value functions $\{V_t^i(\cdot) : t \in \mathcal{T}\}$ by using simple scalar functions. We observe that the optimality equation in (27) has to keep track of the "identities" of the reservations so that the penalty cost given by the optimal objective value of problem (28)-(31) can be computed properly. On the other hand, if we assume that knowing the total number of reservations is adequate to compute the penalty cost, then the state variable in the optimality equation in (27) collapses to a scalar. Our approximation builds on this observation and it is based on approximating the expected penalty cost at the departure time by using only the total number of reservations.

We begin by introducing some new notation. We use $\mathcal{A}^i(\cdot)$ to denote the operator that adds up the components of a $|\mathcal{J}|$ dimensional vector corresponding to the elements of the set $\mathcal{J}^i$. For example, we have $\mathcal{A}^i(x_t) = \sum_{j \in \mathcal{J}^i} x_{jt}$ and $\mathcal{A}^i(x_t)$ is the total number of reservations at the beginning of time period t for the itineraries that use flight leg i. Our approximation is based on the assumption that if we have a total of $\mathcal{A}^i(x_0)$ reservations at the beginning of time period 0 for the itineraries that use flight leg i, then a fixed portion, say α_j^i, of these reservations are for itinerary j. In this case, recalling that the random variable $S_j(\cdot)$ captures the number of reservations for itinerary j that show up at the departure time and defining the vectors $\alpha^i = \{\alpha_j^i : j \in \mathcal{J}^i\}$, $\alpha^i \, \mathcal{A}^i(x_t) = \{\alpha_j^i \, \mathcal{A}^i(x_t) : j \in \mathcal{J}^i\}$ and $S^i(\alpha^i \, \mathcal{A}^i(x_0)) = \{S_j(\alpha_j^i \, \mathcal{A}^i(x_0)) : j \in \mathcal{J}^i\}$, we can approximate the penalty cost at the departure time by $\Gamma^i(S^i(\alpha^i \, \mathcal{A}^i(x_0)))$. In this expression, the vector $\alpha^i \, \mathcal{A}^i(x_0)$ approximates the numbers of reservations that we have at the beginning of time period 0, whereas the vector $S^i(\alpha^i \, \mathcal{A}^i(x_0))$ gives the numbers of reservations that show up at the departure time. The function $\Gamma^i(\cdot)$ is given by the optimal objective value of problem (28)-(31) and it computes the penalty cost for the single leg revenue management problem that takes place over flight leg i. This approximation to the penalty cost at the departure time, in turn, allows us to approximate the solution to the optimality equation in (27) by using the solution to the optimality equation

$$v_t^i(\mathcal{A}^i(x_t)) = \sum_{j \in \mathcal{J}^i} p_{jt} \max\{F_j^i + v_{t-1}^i(\mathcal{A}^i(x_t + e_j)), v_{t-1}^i(\mathcal{A}^i(x_t))\} + \Big[1 - \sum_{j \in \mathcal{J}^i} p_{jt}\Big] v_{t-1}^i(\mathcal{A}^i(x_t)) \qquad (35)$$

with the boundary condition that $v_0^i(\mathcal{A}^i(x_0)) = -\mathbb{E}\{\Gamma^i(S^i(\alpha^i \, \mathcal{A}^i(x_0)))\}$. We note that the optimality equation above involves a scalar state variable and it can be solved quite efficiently.

There are three issues that need to be resolved to be able to find a numerical solution to the optimality equation in (35). The first issue is related to the choice of α_j^i. We use the DLP policy in (12) for this purpose. In particular, we simulate the trajectory of the DLP policy under M itinerary request realizations. Letting $\{x_{jt}^m : j \in \mathcal{J}, \ t \in \mathcal{T}\}$ be the state trajectory in the mth itinerary request realization, we let

$$\alpha_j^i = \frac{\sum_{m=1}^M x_{j0}^m}{\sum_{m=1}^M \sum_{\tilde{j} \in \mathcal{J}^i} x_{\tilde{j}0}^m}.$$

In practice, it is common to use the DLP policy to come up with an average probability that a reservation shows up at the departure time. Our choice of α_j^i closely follows this approach.

The second issue arises due to the fact that the argument of $S_j(\cdot)$ in the vector $S^i(\alpha^i \, \mathcal{A}^i(x_0)) = \{S_j(\alpha_j^i \, \mathcal{A}^i(x_0)) : j \in \mathcal{J}^i\}$ is not necessarily integer. We recall that $S_j(x_{j0})$ is a binomially distributed random variable with parameters (x_{j0}, q_j), but a binomially distributed random variable with a fractional trial parameter is ill-defined. We overcome this issue by always visualizing $S_j(x_{j0})$ as a mixture of two binomially distributed random variables. In particular, letting $\lfloor \cdot \rfloor$ be the round down function, with probability $\lfloor x_{j0} \rfloor + 1 - x_{j0}$, $S_j(x_{j0})$ is equal to a binomially distributed random variable with parameters $(\lfloor x_{j0} \rfloor, q_j)$, and with probability $x_{j0} - \lfloor x_{j0} \rfloor$, $S_j(x_{j0})$ is equal to a binomially distributed random variable with parameters $(\lfloor x_{j0} \rfloor + 1, q_j)$. With this convention, if x_{j0} is integer, then $S_j(x_{j0})$ continues to be binomially distributed with parameters (x_{j0}, q_j). If, however, x_{j0} is fractional, then $S_j(x_{j0})$ is not necessarily binomially distributed, but its expected value continues to be $(\lfloor x_{j0} \rfloor + 1 - x_{j0}) \, q_j \, \lfloor x_{j0} \rfloor + (x_{j0} - \lfloor x_{j0} \rfloor) \, q_j \, (\lfloor x_{j0} \rfloor + 1) = q_j \, x_{j0}$.

Finally, the third issue becomes apparent when we note that the boundary condition of the optimality equation in (35) requires computing the expectation $\mathbb{E}\{\Gamma^i(S^i(\alpha^i \, \mathcal{A}^i(x_0)))\}$ over the multi dimensional random variable $S^i(\alpha^i \, \mathcal{A}^i(x_0))$. There is no closed form expression for this expectation and we simply approximate it through Monte Carlo samples.

Once we agree on the resolution of the three issues described above, we can obtain $\{v_t^i(\cdot) : i \in \mathcal{L}, \ t \in \mathcal{T}\}$ through the optimality equation in (35) and use $\{v_t^i(\cdot) : i \in \mathcal{L}, \ t \in \mathcal{T}\}$ as approximations to $\{V_t^i(\cdot) : i \in \mathcal{L}, \ t \in \mathcal{T}\}$ in the decision rule in (34). In particular, if the state of the reservations at the beginning of time period t is given by x_t, then we accept a request for itinerary j whenever we have

$$f_j \geq \frac{1}{|\mathcal{L}^j|} \sum_{i \in \mathcal{L}^j} \Big\{ v_{t-1}^i(\mathcal{A}^i(x_t)) - v_{t-1}^i(\mathcal{A}^i(x_t + e_j)) + q_j \, \Lambda_j^i \Big\}. \qquad (36)$$

We refer to this decision rule as DPD policy, standing for dynamic programming decomposition.

5 Computational Experiments

In this section, we compare the performances of the decision rules in (12) and (36), along with other benchmark strategies. We begin by describing the experimental setup and the benchmark strategies. Following this, we present our computational results.

5.1 Experimental Setup

We consider an airline network that consists of a hub and N spokes. This is a key network structure that frequently arises in practice. There are two flight legs associated with each spoke. One of these is from the hub to the spoke, whereas the other one is from the spoke to the hub. The airline offers a high fare and a low fare itinerary associated with each origin destination pair. Therefore, the number of flight legs is $2N$ and the number of itineraries is $2N\,(N+1)$. The fare associated with a high fare itinerary is κ times the fare associated with the corresponding low fare itinerary. The penalty cost of denying boarding to a reservation for itinerary j is given by $\theta_j = \delta\, f_j + \sigma\, \max\{f_{\tilde{\jmath}} : \tilde{\jmath} \in \mathcal{J}\}$, where δ and σ are two parameters that we change. The probability that a reservation shows up at the departure time is q and it does not depend on the itinerary. Noting that the total expected demand for the capacity on flight leg i is given by $q \sum_{t\in\mathcal{T}} \sum_{j\in\mathcal{J}} a_{ij}\, p_{jt}$, we measure the tightness of the leg capacities by

$$\rho = \frac{q \sum_{i\in\mathcal{L}} \sum_{t\in\mathcal{T}} \sum_{j\in\mathcal{J}} a_{ij}\, p_{jt}}{\sum_{i\in\mathcal{L}} c_i}.$$

We label our test problems by $(N, \kappa, \delta, \sigma, q, \rho)$ and use $N \in \{4, 8\}$, $\kappa \in \{4, 8\}$, $(\delta, \sigma) \in \{(4, 0), (8, 0), (1, 1)\}$, $q \in \{0.90, 0.95\}$ and $\rho \in \{1.2, 1.6\}$. This provides 48 test problems for our experimental setup. In all of our test problems, we have $\tau = 240$. The online supplement provides the data files for all of our test problems. We describe the format of the data files in Appendix C.

It is worthwhile to note that the interaction between κ and (δ, σ) creates interesting situations. For example, when we have $\kappa = 8$ and $(\delta, \sigma) = (4, 0)$, if the revenue associated with a low fare itinerary is f, then the penalty cost associated with this itinerary is $4f$ and the revenue associated with the corresponding high fare itinerary is $8f$. In this case, if we have a request for the high fare itinerary and a flight leg in this itinerary is already overbooked with a reservation for the low fare itinerary, then we can still accept the high fare itinerary request and deny boarding to the low fare reservation to make a net profit of $8f - 4f$. This corresponds to the case where a high fare itinerary trivially preempts the corresponding low fare itinerary. On the other hand, when we have $\kappa = 4$ and $(\delta, \sigma) = (1, 1)$, such preemptions do not occur. We also note that the test problems with $(\delta, \sigma) = (1, 1)$ tend to have higher penalty costs than the test problems with $(\delta, \sigma) = (8, 0)$, which, in turn, tend to have higher penalty costs than the test problems with $(\delta, \sigma) = (4, 0)$.

5.2 Benchmark Strategies

We compare the performances of the following seven benchmark strategies.

Dynamic programming decomposition (DPD) This benchmark strategy corresponds to the DPD policy given by (36). We use $M = 100$ when computing $\{\alpha^i_j : i \in \mathcal{L},\ j \in \mathcal{J}\}$. We estimate all expectations through 1,000 Monte Carlo samples. With these settings, the 95% confidence interval for the expectation of α_j^i has precision $\mp 4.1\%$ on the average, whereas the 95% confidence interval for the expected penalty cost incurred at the departure time has precision $\mp 1.8\%$ on the average.

Deterministic linear program (DLP) This benchmark strategy corresponds to the DLP policy in (12). The basic variant of this strategy simply solves problem (7)-(11) to obtain the optimal values of the dual variables associated with constraints (8) and uses these dual variable to implement the DLP policy. We use a reoptimized variant of this strategy, where we divide the planning horizon into K equal segments and resolve an updated version of problem (7)-(11) for each segment. In particular, given that the state of the reservations at the beginning of the kth segment is $x_{\tau(K-k+1)/K}$, we replace the right hand side of constraints (8) with $\{c_i - \sum_{j \in \mathcal{J}} a_{ij}\, q_j\, x_{j,\tau(K-k+1)/K} : i \in \mathcal{L}\}$, the right hand side of constraints (9) with $\{\sum_{t=1}^{\tau(K-k+1)/K} p_{jt} : j \in \mathcal{J}\}$ and the right hand side of constraints (10) with $\{q_j\, x_{j,\tau(K-k+1)/K} : j \in \mathcal{J}\}$, and solve this modified version of problem (7)-(11). Letting $\{\lambda_i^* : i \in \mathcal{L}\}$ be the optimal values of the dual variables associated with constraints (8), we use these updated values in the decision rule in (12) until we resolve problem (7)-(11) at the beginning of the next segment. We use $K = 20$ in our computational experiments.

Finite differences in the deterministic linear program (FDD) Given that the state of the reservations at the beginning of time period t is x_t, FDD approximates the optimal total expected profit over the time periods $\{t, \ldots, 0\}$ by using the optimal objective value of the problem

$$\begin{aligned}
\max \quad & \sum_{j \in \mathcal{J}} f_j\, z_j - \sum_{j \in \mathcal{J}} \theta_j\, w_j \\
\text{subject to} \quad & \sum_{j \in \mathcal{J}} a_{ij}\, [q_j\, z_j - w_j] \leq c_i - \sum_{j \in \mathcal{J}} a_{ij}\, q_j\, x_{jt} && i \in \mathcal{L} \\
& z_j \leq \sum_{\tilde{t}=1}^{t} p_{j\tilde{t}} && j \in \mathcal{J} \\
& w_j - q_j\, z_j \leq q_j\, x_{jt} && j \in \mathcal{J} \\
& z_j,\, w_j \geq 0 && j \in \mathcal{J}.
\end{aligned}$$

Thus, letting $L_t(x_t)$ be the optimal objective value of the problem above, FDD uses $L_t(x_t)$ as an approximation to $V_t(x_t)$. In this case, we can make the decisions by replacing $\{V_t(\cdot) : t \in \mathcal{T}\}$ in the decision rule in (6) with $\{L_t(\cdot) : t \in \mathcal{T}\}$. This approach is proposed in Bertsimas and Popescu (2003).

Similar to DLP, we use a reoptimized version of FDD, where we divide the planning horizon into K equal segments and retune the decision rule at the beginning of each segment. Given that the state of the reservations at the beginning of the kth segment is $x_{\tau(K-k+1)/K}$, we compute $L_{\tau(K-k+1)/K}(x_{\tau(K-k+1)/K})$ $- L_{\tau(K-k+1)/K}(x_{\tau(K-k+1)/K} + e_j)$ for all $j \in \mathcal{J}$. Following the decision rule in (6), if we have

$$f_j \geq L_{\tau(K-k+1)/K}(x_{\tau(K-k+1)/K}) - L_{\tau(K-k+1)/K}(x_{\tau(K-k+1)/K} + e_j)$$

then we always accept a request for itinerary j until we reach the beginning of the next segment and retune the decision rule. We use $K = 20$.

Virtual capacities based on a naive computation (VCN) In this benchmark strategy, the airline sets virtual capacities on the flight legs by assuming that the no shows take on their expected values. Following this, the airline makes the capacity allocation decisions under the assumption that all of the reservations show up, but the capacities on the flight legs are equal to the virtual capacities. In other words, noting that a reservation shows up at the departure time with probability q, the airline sets the virtual capacity on flight leg i as $u_i = \lfloor c_i/q \rfloor$ and solves a version of the deterministic linear program in (7)-(11), which can be stated as

$$\max \quad \sum_{j \in \mathcal{J}} f_j \, z_j \tag{37}$$

$$\text{subject to} \quad \sum_{j \in \mathcal{J}} a_{ij} \, z_j \leq u_i \qquad i \in \mathcal{L} \tag{38}$$

$$z_j \leq \sum_{t \in \mathcal{T}} p_{jt} \qquad j \in \mathcal{J} \tag{39}$$

$$z_j \geq 0 \qquad j \in \mathcal{J}. \tag{40}$$

Letting $\{\lambda_i^* : i \in \mathcal{L}\}$ be the optimal values of the dual variables associated with the first set of constraints above, VCN uses the DLP policy in (12). Similar to DLP and FDD, we use a reoptimized version of VCN with 20 reoptimizations.

Virtual capacities based on an economic model (VCE) One criticism for VCN is that it chooses the virtual capacities under the assumption that the no shows take on their expected values. However, depending on the tradeoffs between the fares, penalty costs and show up probabilities, we may want to be more or less aggressive than what the expected values of the no shows suggest. The goal of VCE is to make up for this shortcoming. VCE is proposed in Karaesmen and van Ryzin (2004*a*) and it is based on the following three assumptions. First, the revenue that we make from one unit of capacity on a flight leg is known. We let r_i be the revenue that we make from one unit of capacity on flight leg i. Second, if a reservation uses the capacities on multiple flight legs, then we can allow boarding to this reservation on one flight leg, while denying boarding to the same reservation on another flight leg. Furthermore, the penalty cost of denying boarding to a reservation on a flight leg is known. We let g_i be the penalty cost of denying boarding to a reservation on flight leg i. Third, if the airline sets the virtual capacity on flight leg i as u_i, then it sells exactly u_i reservations on flight leg i.

By the third assumption, if we set the virtual capacity on flight leg i as u_i, then we sell u_i reservations on flight leg i, in which case, the first assumption implies that we generate a revenue of $r_i \, u_i$. On the other hand, if we let $B_i(u_i)$ be a binomially distributed random variable with parameters (u_i, q), then the number of reservations that show up at the departure time is given by $B_i(u_i)$ and the second assumption implies that the expected penalty cost that we incur on flight leg i is $g_i \, \mathbb{E}\{\max\{B_i(u_i) - c_i, 0\}\}$. Therefore, VCE solves the problem $\max_{u_i} r_i \, u_i - g_i \, \mathbb{E}\{\max\{B_i(u_i) - c_i, 0\}\}$ to set the virtual capacity on flight leg i. Once the virtual capacities have been set, VCE proceeds in the same way as VCN.

Karaesmen and van Ryzin (2004*a*) suggest several choices for r_i and g_i. Following their work, we let $R_j = f_j / \sum_{l \in \mathcal{L}} a_{lj}$ and $G_j = \theta_j / \sum_{l \in \mathcal{L}} a_{lj}$ for all $j \in \mathcal{J}$ to evenly distribute the revenue and penalty cost associated with an itinerary over the flight legs that it uses. In this case, we try choosing r_i as

$r_i = \sum_{j \in \mathcal{J}^i} R_j / |\mathcal{J}^i|$ or $r_i = \max\{R_j : j \in \mathcal{J}^i\}$ or $r_i = \min\{R_j : j \in \mathcal{J}^i\}$, and g_i as $g_i = \sum_{j \in \mathcal{J}^i} G_j / |\mathcal{J}^i|$ or $g_i = \max\{G_j : j \in \mathcal{J}^i\}$ or $g_i = \min\{G_j : j \in \mathcal{J}^i\}$. Using all combinations of these choices, we have nine different choices for r_i and g_i. We test the performances of all of these nine choices for all of our test problems, but for brevity, only report the results corresponding to the best choice. For different test problems, the best choice for r_i and g_i can be different. Similar to VCN, we use a reoptimized version of VCE with 20 reoptimizations.

Virtual capacities joint with capacity allocation decisions (VCJ) Both VCN and VCE use the assumption that we can set the virtual capacities first, and then, come up with a policy to accept or reject the itinerary requests. In contrast, VCJ uses the penalty cost $g_i \, \mathbb{E}\{\max\{B_i(u_i) - c_i, 0\}\}$ in problem (37)-(40) to jointly set the virtual capacities and come up with a policy to accept or reject the itinerary requests. In particular, VCJ solves the problem

$$
\begin{aligned}
\max \quad & \sum_{j \in \mathcal{J}} f_j \, z_j - \sum_{i \in \mathcal{L}} g_i \, \mathbb{E}\{\max\{B_i(u_i) - c_i, 0\}\} \\
\text{subject to} \quad & \sum_{j \in \mathcal{J}} a_{ij} \, z_j - u_i \leq 0 \qquad i \in \mathcal{L} \\
& z_j \leq \sum_{t \in \mathcal{T}} p_{jt} \qquad j \in \mathcal{J} \\
& z_j, u_i \geq 0 \qquad i \in \mathcal{L}, \; j \in \mathcal{J},
\end{aligned}
$$

where we use interpolations of the function $\mathbb{E}\{\max\{B_i(u_i) - c_i, 0\}\}$ to be able to compute it at a fractional u_i. Letting $\{\lambda_i^* : i \in \mathcal{L}\}$ be the optimal values of the dual variables associated with the first set of constraints above, VCJ uses the DLP policy in (12). This approach is proposed in Karaesmen and van Ryzin (2004*a*). Similar to VCE, we try three different choices for g_i and report the results corresponding to the best choice. We use a reoptimized version of VCJ with 20 reoptimizations.

Separable penalty costs (SPC) This benchmark strategy is developed by Erdelyi and Topaloglu (2009). The fundamental observation behind SPC is that if the penalty cost of denying boarding to the reservations were given by a separable function of the form $\Gamma(S(x_0)) = \sum_{j \in \mathcal{J}} \gamma_j(S_j(x_{j0}))$, then the optimality equation in (5) would decompose by the itineraries. To exploit this observation, SPC approximates $\Gamma(S(x_0))$ in problem (1)-(4) with a separable function of the form $\sum_{j \in \mathcal{J}} \gamma_j(S_j(x_{j0}))$ and solves the optimality equation in (5) with the approximate boundary condition that $V_0(x_0) = -\mathbb{E}\{\sum_{j \in \mathcal{J}} \gamma_j(S_j(x_{j0}))\}$. The value functions $\{V_t(\cdot) : t \in \mathcal{T}\}$ obtained in this fashion are used to construct a policy to accept or reject the itinerary requests. SPC uses a simulation based method to construct the separable approximation $\sum_{j \in \mathcal{J}} \gamma_j(S_j(x_{j0}))$ to the penalty cost. Roughly speaking, we simulate the DLP policy in (12) to have a general idea about the numbers of reservations that show up at the departure time. Following this, we compute the slopes of $\Gamma(\cdot)$ at these numbers of reservations along different directions and use these slopes to construct the scalar functions $\{\gamma_j(\cdot) : j \in \mathcal{J}\}$. An exact description of this benchmark strategy is beyond the scope of our paper and we refer the reader to Erdelyi and Topaloglu (2009) for the details. Similar to the other benchmark strategies, we retune the separable approximation five times over the planning horizon. It turns out that retuning SPC more than five times does not provide any additional benefit.

5.3 Computational Results

Our main computational results are summarized in Tables 1 and 2. In particular, these two tables respectively show the results for the test problems with four and eight spokes. The first column in Tables 1 and 2 gives the characteristics of the test problem. The second column gives the upper bound on the optimal total expected profit provided by the optimal objective value of problem (7)-(11). The next seven columns give the total expected profits obtained by DPD, DLP, FDD, VCN, VCE, VCJ and SPC. These total expected profits are estimated by simulating the performances of the different policies under 50 itinerary request trajectories. We use common itinerary request trajectories when simulating the performances of the different policies. The tenth column gives the percent gap between the total expected profits obtained by DPD and DLP. This column also includes a "✓" whenever DPD performs significantly better than DLP and a " $\odot$" whenever there is no statistically significant performance gap between the two methods at 95% level. The last five columns do the same thing as the tenth column, but they compare the performance of DPD with FDD, VCN, VCE, VCJ and SPC.

The results indicate that DPD performs substantially better than all of the benchmark strategies that use a linear programming formulation. Among the linear programming based benchmark strategies, FDD performs the best and it is followed by VCJ, DLP, VCE and VCN. The superiority of FDD over DLP is also observed by Bertsimas and Popescu (2003). The performance gaps between DPD and DLP, FDD, VCN, VCE and VCJ are statistically significant for all of the test problems. The average performance gaps between DPD and DLP, FDD, VCN, VCE and VCJ are respectively 4.09, 2.85 6.82, 4.76 and 2.82 for the test problems with four spokes. The same gaps increase to 5.39, 2.87, 8.18, 6.08 and 3.53 for the test problems with eight spokes. Among the three benchmark strategies that use virtual capacities, VCJ performs better than VCN and VCE. It is interesting to note that VCJ performs better than DLP as well. There are test problems where VCN performs better than DLP, despite the fact that VCN is essentially an ad hoc modification of DLP that does not carefully address the possibility of no shows. However, the performance of VCN is not robust as indicated by the test problems with $(\delta, \sigma) = (4, 0)$. There is not a clear distinction between DLP and VCE, but there are test problems where VCE can perform substantially worse than DLP.

The performance gap between DPD and SPC is on the order of half a percent to a percent. We emphasize that a percent revenue difference is still considered significant in the revenue management setting. On a majority of the test problems, DPD performs better than SPC and in the remaining test problems, there does not exist a statistically significant gap between the two benchmark strategies. Similar to DPD, SPC performs noticeably better than all of the benchmark strategies that use a linear programming formulation. Therefore, DPD and SPC, by working with the dynamic programming formulation of the capacity allocation and overbooking problem, provide significant improvements over using a deterministic linear programming formulation, which ignores the temporal dynamics of the arrivals of the itinerary requests.

It is possible to observe a few trends in the performance gaps. In particular, the performance gaps between DPD and the linear programming based benchmark strategies tend to increase as the fare difference between the high fare and low fare itineraries, penalty costs and overall tightness of the leg capacities increase. For test problems with large fare differences, large penalty costs and tight leg capacities, the "regret" associated with making an "incorrect" decision is relatively large. For example, when the fare difference between the high fare and low fare itineraries is large, accepting a request for a low fare itinerary "incorrectly" may preclude accepting a request for a high fare itinerary later in the planning horizon and the revenue forgone in this case can be quite large. Similarly, when the penalty costs are large, it is costly to deny boarding to a reservation that was accepted "by mistake." When the leg capacities are tight, it is important to make the itinerary acceptance decisions more "carefully," as it is not possible to accommodate all itinerary requests. Thus, it is encouraging that a careful stochastic model pays off and DPD performs significantly better than the linear programming based benchmark strategies as the fare differences, penalty costs and tightness of the leg capacities increase. To display some of these trends, Table 3 shows the performance gaps between DPD and the other benchmark strategies averaged over a number of test problems with a particular characteristic. For example, the second column shows the performance gaps averaged over the test problems with four spokes. The trends that we mention can be observed from this table.

Table 4 shows the CPU seconds required to compute one set of value function approximations for DPD and SPC. All of the computational experiments are run on a Pentium IV desktop PC with 2.4 GHz CPU and 1 GB RAM. Since the number of spokes appears to be the primary factor affecting the computation times, we give the average CPU seconds over different test problems. The two rows in Table 4 show the CPU seconds for DPD and SPC. The second and third columns respectively correspond to the test problems with four and eight spokes. The CPU seconds for DPD includes the operations required to estimate $\{\alpha_j^i : i \in \mathcal{L},\ j \in \mathcal{J}\}$ and to compute $\{v_t^i(\cdot) : i \in \mathcal{L},\ t \in \mathcal{T}\}$. The results indicate that DPD takes significantly less time than SPC and scales more favorably. Considering its performance, DPD appears to be preferable to SPC. DLP, FDD, VCN, VCE and VCJ take at most a few seconds to reoptimize their decision rules. Despite this extra computational burden, the computational requirement for DPD is still reasonable. Given the substantial improvements that it provides over the other benchmark strategies, DPD appears to be a viable choice.

6 Conclusions

In this paper, we developed a network revenue management model to jointly make the capacity allocation and overbooking decisions over an airline network. Our approach is based on decomposing the network revenue management problem into a sequence of single leg revenue management problems and exploiting the observation that if the proportions of the reservations at the departure time were known, then the dynamic programming formulation of the single leg revenue management problems would involve only a scalar state variable. Using these observations, we constructed tractable approximations to the value functions. Computational experiments demonstrated that the resulting policies perform significantly better than the benchmark strategies.

Acknowledgements

The authors thank two anonymous referees for their useful comments that especially strengthened the exposition and computational results. This work was supported in part by National Science Foundation grants DMI-0422133 and CMMI-0758441.

References

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

Beckmann, M. J. (1958), 'Decision and team problems in airline reservations', *Econometrica* **26**(1), 134–145.

Belobaba, P. P. (1987), Air Travel Demand and Airline Seat Inventory Control, PhD thesis, Massachusetts Institute of Technology, Cambridge, MA.

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

Bront, J. J. M., Mendez-Diaz, I. and Vulcano, G. (2008), 'A column generation algorithm for choice-based network revenue management', *Operations Research* (to appear).

Chatwin, R. E. (1992), 'Multiperiod airline overbooking with a single fare class', *Operations Research* **46**(6), 805–819.

Chatwin, R. E. (1999), 'Continuous-time airline overbooking with time-dependent fares and refunds', *Transportation Science* **33**(2), 182–191.

Coughlan, J. (1999), 'Airline overbooking in the multi-class case', *The Journal of the Operational Research Society* **50**(11), 1098–1103.

Erdelyi, A. and Topaloglu, H. (2009), 'Separable approximations for joint capacity control and overbooking decisions in network revenue management', *Journal of Revenue and Pricing Management* **8**(1), 3–20.

Gallego, G. and van Ryzin, G. (1997), 'A multiproduct dynamic pricing problem and its applications to yield management', *Operations Research* **45**(1), 24–41.

Karaesmen, I. and van Ryzin, G. (2004*a*), Coordinating overbooking and capacity control decisions on a network, Technical report, Columbia Business School.

Karaesmen, I. and van Ryzin, G. (2004*b*), 'Overbooking with substitutable inventory classes', *Operations Research* **52**(1), 83–104.

Kleywegt, A. J. (2001), An optimal control problem of dynamic pricing, Technical report, School of Industrial and Systems Engineering, Georgia Institute of Technology.

Kunnumkal, S. and Topaloglu, H. (2007), A new dynamic programming decomposition method for the network revenue management problem with customer choice behavior, Technical report, Cornell University, School of Operations Research and Information Engineering.
Available at `http://legacy.orie.cornell.edu/~huseyin/publications/publications.html`.

Liu, Q. and van Ryzin, G. (2008), 'On the choice-based linear programming model for network revenue management', *Manufacturing & Service Operations Management* **10**(2), 288–310.

Meissner, J. and Strauss, A. K. (2008), Network revenue management with inventory-sensitive bid prices and customer choice, Technical report, Lancaster University Management School, Department of Management Science,.

Subramanian, J., Stidham, S. and Lautenbacher, C. J. (1999), 'Airline yield management with overbooking, cancellations and no-shows', *Transportation Science* **33**(2), 147–167.

Talluri, K. T. and van Ryzin, G. J. (2004), *The Theory and Practice of Revenue Management*, Kluwer Academic Publishers.

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

Thompson, H. R. (1961), 'Statistical problems in airline reservation control', *Journal of the Operational Research Society* **12**(3), 167–185.

Topaloglu, H. (2006), 'Using Lagrangian relaxation to compute capacity-dependent bid-prices in network revenue management', *Operations Research* (to appear).

Williamson, E. L. (1992), Airline Network Seat Control, PhD thesis, Massachusetts Institute of Technology, Cambridge, MA.

Zhang, D. and Adelman, D. (2006), An approximate dynamic programming approach to network revenue management with customer choice, Technical report, University of Chicago, Graduate School of Business.

Problem	Profit	Total expected profit obtained by							DPD vs.					
$(N,\kappa,\delta,\sigma,q,\rho)$	bound	DPD	DLP	FDD	VCN	VCE	VCJ	SPC	DLP	FDD	VCN	VCE	VCJ	SPC
(4, 4, 4, 0, 0.90, 1.2)	15,223	14,575	14,283	14,386	13,671	14,158	14,359	14,442	2.00 ✓	1.30 ✓	6.21 ✓	2.87 ✓	1.48 ✓	0.91 ✓
(4, 4, 4, 0, 0.90, 1.6)	20,997	19,618	19,085	19,395	18,164	18,716	19,237	19,506	2.72 ✓	1.14 ✓	7.41 ✓	4.60 ✓	1.94 ✓	0.57 ✓
(4, 4, 4, 0, 0.95, 1.2)	23,450	22,108	21,962	21,998	21,510	21,734	21,904	22,062	0.66 ✓	0.50 ✓	2.71 ✓	1.69 ✓	0.92 ✓	0.21 ⊙
(4, 4, 4, 0, 0.95, 1.6)	21,753	20,647	20,208	20,373	19,645	19,873	20,088	20,532	2.13 ✓	1.33 ✓	4.85 ✓	3.75 ✓	2.71 ✓	0.56 ✓
(4, 4, 8, 0, 0.90, 1.2)	23,136	21,672	21,014	21,144	20,695	21,103	21,359	21,487	3.03 ✓	2.43 ✓	4.50 ✓	2.62 ✓	1.44 ✓	0.85 ✓
(4, 4, 8, 0, 0.90, 1.6)	12,177	11,157	10,326	10,620	10,350	10,649	10,741	11,047	7.45 ✓	4.81 ✓	7.24 ✓	4.55 ✓	3.73 ✓	0.99 ✓
(4, 4, 8, 0, 0.95, 1.2)	19,206	17,883	17,285	17,491	17,291	17,431	17,582	17,838	3.35 ✓	2.19 ✓	3.31 ✓	2.53 ✓	1.68 ✓	0.25 ⊙
(4, 4, 8, 0, 0.95, 1.6)	15,995	14,808	14,059	14,316	13,999	14,119	14,198	14,526	5.06 ✓	3.32 ✓	5.46 ✓	4.66 ✓	4.12 ✓	1.90 ✓
(4, 4, 1, 1, 0.90, 1.2)	18,418	16,970	16,362	16,529	16,241	16,543	16,667	16,894	3.58 ✓	2.60 ✓	4.30 ✓	2.52 ✓	1.79 ✓	0.45 ⊙
(4, 4, 1, 1, 0.90, 1.6)	10,626	9,813	9,159	9,372	9,209	9,432	9,506	9,727	6.66 ✓	4.50 ✓	6.16 ✓	3.89 ✓	3.13 ✓	0.88 ✓
(4, 4, 1, 1, 0.95, 1.2)	19,782	18,538	17,797	17,968	17,731	17,897	18,087	18,332	4.00 ✓	3.08 ✓	4.35 ✓	3.46 ✓	2.43 ✓	1.11 ✓
(4, 4, 1, 1, 0.95, 1.6)	17,345	16,072	15,264	15,522	15,210	15,408	15,627	16,019	5.03 ✓	3.42 ✓	5.36 ✓	4.13 ✓	2.77 ✓	0.33 ⊙
(4, 8, 4, 0, 0.90, 1.2)	30,754	29,514	29,286	29,329	26,911	28,065	29,217	29,445	0.77 ✓	0.63 ✓	8.82 ✓	4.91 ✓	1.01 ✓	0.23 ✓
(4, 8, 4, 0, 0.90, 1.6)	31,744	30,841	30,324	30,483	27,204	28,548	30,187	30,679	1.68 ✓	1.16 ✓	11.79 ✓	7.43 ✓	2.12 ✓	0.52 ✓
(4, 8, 4, 0, 0.95, 1.2)	28,983	27,676	27,386	27,445	25,854	26,201	27,160	27,533	1.05 ✓	0.83 ✓	6.58 ✓	5.33 ✓	1.86 ✓	0.51 ✓
(4, 8, 4, 0, 0.95, 1.6)	23,995	22,983	22,720	22,825	21,200	21,616	22,322	22,901	1.14 ✓	0.69 ✓	7.76 ✓	5.95 ✓	2.88 ✓	0.36 ✓
(4, 8, 8, 0, 0.90, 1.2)	26,932	25,888	25,115	25,182	23,503	24,376	25,198	25,825	2.98 ✓	2.72 ✓	9.21 ✓	5.84 ✓	2.66 ✓	0.24 ⊙
(4, 8, 8, 0, 0.90, 1.6)	30,670	28,617	27,314	27,731	25,859	26,718	27,910	28,527	4.55 ✓	3.10 ✓	9.64 ✓	6.64 ✓	2.47 ✓	0.31 ⊙
(4, 8, 8, 0, 0.95, 1.2)	33,136	31,787	31,134	31,267	30,054	30,551	30,989	31,816	2.05 ✓	1.64 ✓	5.45 ✓	3.89 ✓	2.51 ✓	-0.09 ⊙
(4, 8, 8, 0, 0.95, 1.6)	27,926	26,747	25,456	25,781	24,482	24,712	24,931	26,533	4.83 ✓	3.61 ✓	8.47 ✓	7.61 ✓	6.79 ✓	0.80 ✓
(4, 8, 1, 1, 0.90, 1.2)	26,673	25,187	23,050	23,446	23,238	24,046	24,446	25,114	8.48 ✓	6.92 ✓	7.74 ✓	4.53 ✓	2.94 ✓	0.29 ⊙
(4, 8, 1, 1, 0.90, 1.6)	31,470	29,730	27,215	28,178	26,748	27,776	28,623	29,498	8.46 ✓	5.22 ✓	10.03 ✓	6.57 ✓	3.72 ✓	0.78 ✓
(4, 8, 1, 1, 0.95, 1.2)	21,959	20,858	19,411	19,832	19,506	19,799	20,062	20,668	6.94 ✓	4.92 ✓	6.48 ✓	5.08 ✓	3.82 ✓	0.91 ✓
(4, 8, 1, 1, 0.95, 1.6)	26,138	24,341	22,010	22,825	21,959	22,105	22,423	24,040	9.58 ✓	6.23 ✓	9.79 ✓	9.19 ✓	7.88 ✓	1.24 ✓
Average									4.09	2.85	6.82	4.76	2.87	0.63

Table 1: Computational results for the test problems with four spokes.

Problem $(N,\kappa,\delta,\sigma,q,\rho)$	Profit bound	Total expected profit obtained by DPD	DLP	FDD	VCN	VCE	VCJ	SPC	DPD vs. DLP	FDD	VCN	VCE	VCJ	SPC
$(8,4,4,0,0.90,1.2)$	22,706	20,671	20,338	20,463	19,114	19,773	20,428	20,426	1.61 ✓	1.00 ✓	7.53 ✓	4.34 ✓	1.17 ✓	1.18 ✓
$(8,4,4,0,0.90,1.6)$	17,715	15,951	15,555	15,783	14,491	15,124	15,647	15,823	2.48 ✓	1.05 ✓	9.15 ✓	5.19 ✓	1.90 ✓	0.80 ✓
$(8,4,4,0,0.95,1.2)$	21,809	19,960	19,640	19,795	18,903	19,094	19,596	19,829	1.60 ✓	0.83 ✓	5.29 ✓	4.34 ✓	1.82 ✓	0.65 ✓
$(8,4,4,0,0.95,1.6)$	15,625	14,112	13,771	14,025	13,058	13,080	13,606	13,982	2.41 ✓	0.61 ✓	7.47 ✓	7.31 ✓	3.59 ✓	0.92 ✓
$(8,4,8,0,0.90,1.2)$	19,963	17,747	16,783	17,223	16,529	17,006	17,510	17,603	5.43 ✓	2.95 ✓	6.87 ✓	4.18 ✓	1.34 ✓	0.82 ✓
$(8,4,8,0,0.90,1.6)$	13,868	12,189	11,408	11,740	11,211	11,644	11,854	11,961	6.40 ✓	3.68 ✓	8.02 ✓	4.47 ✓	2.75 ✓	1.87 ✓
$(8,4,8,0,0.95,1.2)$	21,134	18,957	18,203	18,589	18,119	18,239	18,382	18,740	3.98 ✓	1.95 ✓	4.42 ✓	3.79 ✓	3.04 ✓	1.15 ✓
$(8,4,8,0,0.95,1.6)$	17,056	14,658	13,768	14,251	13,904	13,920	14,047	14,484	6.08 ✓	2.78 ✓	5.14 ✓	5.04 ✓	4.17 ✓	1.19 ✓
$(8,4,1,1,0.90,1.2)$	20,019	17,932	16,785	17,274	16,878	17,325	17,546	17,867	6.39 ✓	3.67 ✓	5.88 ✓	3.38 ✓	2.15 ✓	0.36 ⊙
$(8,4,1,1,0.90,1.6)$	15,712	13,638	12,488	13,057	12,440	12,902	13,113	13,498	8.43 ✓	4.26 ✓	8.78 ✓	5.39 ✓	3.85 ✓	1.03 ✓
$(8,4,1,1,0.95,1.2)$	16,179	14,145	13,427	13,816	13,637	13,807	13,891	14,096	5.08 ✓	2.32 ✓	3.59 ✓	2.39 ✓	1.79 ✓	0.35 ⊙
$(8,4,1,1,0.95,1.6)$	19,803	17,255	16,119	16,792	16,431	16,492	16,655	17,171	6.58 ✓	2.68 ✓	4.78 ✓	4.42 ✓	3.48 ✓	0.49 ⊙
$(8,8,4,0,0.90,1.2)$	35,075	33,024	32,742	32,905	29,371	30,806	32,739	32,879	0.85 ✓	0.36 ✓	11.06 ✓	6.71 ✓	0.86 ✓	0.44 ✓
$(8,8,4,0,0.90,1.6)$	24,105	22,528	22,120	22,285	19,609	20,448	22,054	22,336	1.81 ✓	1.08 ✓	12.96 ✓	9.23 ✓	2.10 ✓	0.85 ✓
$(8,8,4,0,0.95,1.2)$	33,872	32,052	31,832	31,954	29,486	29,958	31,401	31,808	0.69 ✓	0.30 ✓	8.00 ✓	6.53 ✓	2.03 ✓	0.76 ✓
$(8,8,4,0,0.95,1.6)$	25,920	24,254	23,844	24,056	22,142	22,256	23,403	24,069	1.69 ✓	0.82 ✓	8.71 ✓	8.24 ✓	3.51 ✓	0.76 ✓
$(8,8,8,0,0.90,1.2)$	31,831	29,377	28,228	28,680	26,432	27,564	28,667	29,046	3.91 ✓	2.37 ✓	10.03 ✓	6.17 ✓	2.42 ✓	1.13 ✓
$(8,8,8,0,0.90,1.6)$	37,769	34,116	32,225	33,149	29,732	31,176	32,782	33,955	5.54 ✓	2.84 ✓	12.85 ✓	8.62 ✓	3.91 ✓	0.47 ⊙
$(8,8,8,0,0.95,1.2)$	28,695	26,540	25,549	25,984	24,661	24,860	25,327	26,245	3.74 ✓	2.09 ✓	7.08 ✓	6.33 ✓	4.57 ✓	1.11 ✓
$(8,8,8,0,0.95,1.6)$	32,840	30,092	28,585	29,292	27,706	27,703	28,119	29,804	5.01 ✓	2.66 ✓	7.93 ✓	7.94 ✓	6.56 ✓	0.96 ✓
$(8,8,1,1,0.90,1.2)$	29,394	26,832	23,601	24,840	24,442	25,266	25,681	26,660	12.04 ✓	7.42 ✓	8.90 ✓	5.84 ✓	4.29 ✓	0.64 ✓
$(8,8,1,1,0.90,1.6)$	28,433	25,863	22,104	23,688	22,464	23,741	24,129	25,534	14.53 ✓	8.41 ✓	13.14 ✓	8.20 ✓	6.71 ✓	1.27 ✓
$(8,8,1,1,0.95,1.2)$	26,884	24,647	22,162	23,190	22,743	22,886	23,096	24,396	10.09 ✓	5.91 ✓	7.73 ✓	7.15 ✓	6.29 ✓	1.02 ✓
$(8,8,1,1,0.95,1.6)$	28,228	25,919	22,559	24,123	23,051	23,117	23,244	25,701	12.96 ✓	6.93 ✓	11.06 ✓	10.81 ✓	10.32 ✓	0.84 ✓
Average									5.39	2.87	8.18	6.08	3.53	0.88

Table 2: Computational results for the test problems with eight spokes.

Benchmark strategies	N		κ		(δ, σ)			ρ	
	4	8	4	8	(4,0)	(8,0)	(1,1)	1.2	1.6
DPD vs. DLP	4.09	5.39	4.26	5.22	1.58	4.59	8.05	3.93	5.55
DPD vs. FDD	2.85	2.87	2.43	3.29	0.85	2.82	4.91	2.54	3.18
DPD vs. VCN	6.82	8.18	5.78	9.22	7.89	7.23	7.38	6.50	8.50
DPD vs. VCE	4.76	6.08	3.98	6.86	5.53	5.30	5.43	4.43	6.41
DPD vs. VCJ	2.87	3.53	2.47	3.93	1.99	3.38	4.21	2.53	4.05
DPD vs. SPC	0.63	0.88	0.83	0.68	0.64	0.87	0.75	0.65	0.86

Table 3: Comparison of the performances of DPD and the other benchmark strategies for different sets of test problems.

Benchmark strategy	N	
	4	8
DPD	48	76
SPC	118	220

Table 4: CPU seconds for DPD and SPC.

A Appendix: Optimal Objective Value of Problem (22)-(25)

Letting $\mathbf{1}(\cdot)$ be the indicator function and noting constraints (24), the optimal values of the decision variables $\{w_j : j \in \mathcal{J} \setminus \mathcal{J}^i\}$ are $\{\mathbf{1}(\Theta^i_j \leq 0)\, q_j\, z_j : j \in \mathcal{J} \setminus \mathcal{J}^i\}$. Therefore, letting $[\cdot]^+ = \max\{\cdot, 0\}$, problem (22)-(25) can be written as

$$\begin{aligned} \max \quad & \sum_{j \in \mathcal{J} \setminus \mathcal{J}^i} \big[F^i_j + [-\Theta^i_j]^+\, q_j\big]\, z_j \\ \text{subject to} \quad & z_j \leq \sum_{t \in \mathcal{T}} p_{jt} && j \in \mathcal{J} \setminus \mathcal{J}^i \\ & z_j \geq 0 && j \in \mathcal{J} \setminus \mathcal{J}^i. \end{aligned}$$

In the problem above, the optimal values of the decision variables $\{z_j : j \in \mathcal{J} \setminus \mathcal{J}^i\}$ are $\{\mathbf{1}([F^i_j + [-\Theta^i_j]^+\, q_j] \geq 0) \sum_{t \in \mathcal{T}} p_{jt} : j \in \mathcal{J} \setminus \mathcal{J}^i\}$. Therefore, the optimal objective value of the problem above is

$$\sum_{t \in \mathcal{T}} \sum_{j \in \mathcal{J} \setminus \mathcal{J}^i} \big[F^i_j + [-\Theta^i_j]^+\, q_j\big]^+ p_{jt} = \sum_{t \in \mathcal{T}} \sum_{j \in \mathcal{J} \setminus \mathcal{J}^i} \max\{F^i_j, F^i_j - q_j\, \Theta^i_j, 0\}\, p_{jt}.$$

B Appendix: Proof of Proposition 2

To simplify the proof, we introduce auxiliary value functions $\{\psi^i_t(\cdot) : i \in \mathcal{L},\ t \in \mathcal{T}\}$ by letting

$$\psi^i_t(x_t) = \sum_{j \in \mathcal{J}} p_{jt} \max\{F^i_j + \psi^i_{t-1}(x_t + e_j), \psi^i_{t-1}(x_t)\} + \Big[1 - \sum_{j \in \mathcal{J}} p_{jt}\Big] \psi^i_{t-1}(x_t) \tag{41}$$

with the boundary condition that $\psi^i_0(x_0) = -\mathbb{E}\{\phi^i(S(x_0))\}$, where

$$\begin{aligned} \phi^i(S(x_0)) = \min \quad & \sum_{j \in \mathcal{J}} \Theta^i_j\, w_j && && (42) \\ \text{subject to} \quad & \sum_{j \in \mathcal{J}} a_{ij}\, [S_j(x_{j0}) - w_j] \leq c_i && && (43) \\ & w_j \leq S_j(x_{j0}) && j \in \mathcal{J} && (44) \\ & w_j \in \mathbb{Z}_+ && j \in \mathcal{J}. && (45) \end{aligned}$$

The following two results provide the intermediate steps to prove Proposition 2.

Lemma 3 *For all $t \in \mathcal{T}$, we have*

$$\psi^i_t(x_t) = V^i_t(\mathcal{R}^i(x_t)) + \sum_{j \in \mathcal{J} \setminus \mathcal{J}^i} q_j\, [-\Theta^i_j]^+\, x_{jt} + \sum_{s=1}^{t} \sum_{j \in \mathcal{J} \setminus \mathcal{J}^i} p_{js} \max\{F^i_j, F^i_j - q_j\, \Theta^i_j, 0\}.$$

Proof of Lemma 3 We show the result by induction over the time periods. Noting the upper bounds on the decision variables $\{w_j : j \in \mathcal{J} \setminus \mathcal{J}^i\}$ in problem (42)-(45), the optimal values of these decision variables are $\{\mathbf{1}(\Theta^i_j \leq 0)\, S_j(x_{j0}) : j \in \mathcal{J} \setminus \mathcal{J}^i\}$. Thus, since we have $a_{ij} = 0$ for all $j \in \mathcal{J} \setminus \mathcal{J}^i$, we have

$$\phi^i(S(x_0)) = \Gamma^i(\mathcal{R}^i(S(x_0))) - \sum_{j \in \mathcal{J} \setminus \mathcal{J}^i} [-\Theta^i_j]^+\, S_j(x_{j0}),$$

where $\Gamma^i(\mathcal{R}^i(S(x_0)))$ is the optimal objective value of problem (28)-(31). Taking expectations in the expression above and noting that $S_j(x_{j0})$ has a binomial distribution with parameters (x_{j0}, q_j), we obtain $\psi^i_0(x_0) = -\mathbb{E}\{\phi^i(S(x_0))\} = -\mathbb{E}\{\Gamma^i(\mathcal{R}^i(S(x_0)))\} + \sum_{j\in\mathcal{J}\setminus\mathcal{J}^i} q_j\,[-\Theta^i_j]^+ x_{j0} = V^i_0(\mathcal{R}^i(x_0)) + \sum_{j\in\mathcal{J}\setminus\mathcal{J}^i} q_j\,[-\Theta^i_j]^+ x_{j0}$ and the result holds for the last time period. Assuming that the result holds for time period $t-1$, we have

$$\psi^i_{t-1}(x_t + e_j) - \psi^i_{t-1}(x_t) = \begin{cases} V^i_{t-1}(\mathcal{R}^i(x_t+e_j)) - V^i_{t-1}(\mathcal{R}^i(x_t)) & \text{if } j \in \mathcal{J}^i \\ q_j\,[-\Theta^i_j]^+ & \text{if } j \in \mathcal{J}\setminus\mathcal{J}^i. \end{cases} \tag{46}$$

Therefore, we have

$$\begin{aligned} \psi^i_t(x_t) &= \sum_{j\in\mathcal{J}} p_{jt} \max\{F^i_j + \psi^i_{t-1}(x_t+e_j) - \psi^i_{t-1}(x_t), 0\} + \psi^i_{t-1}(x_t) \\ &= \sum_{j\in\mathcal{J}^i} p_{jt} \max\{F^i_j + V^i_{t-1}(\mathcal{R}^i(x_t+e_j)) - V^i_{t-1}(\mathcal{R}^i(x_t)), 0\} \\ &\qquad + \sum_{j\in\mathcal{J}\setminus\mathcal{J}^i} p_{jt} \max\{F^i_j + q_j\,[-\Theta^i_j]^+, 0\} + V^i_{t-1}(\mathcal{R}^i(x_t)) \\ &\qquad\qquad + \sum_{j\in\mathcal{J}\setminus\mathcal{J}^i} q_j\,[-\Theta^i_j]^+\, x_{jt} + \sum_{s=1}^{t-1} \sum_{j\in\mathcal{J}\setminus\mathcal{J}^i} p_{js} \max\{F^i_j, F^i_j - q_j\,\Theta^i_j, 0\}, \end{aligned}$$

where the first equality follows from (41) and the second equality follows from (46) and the induction assumption. Since $\max\{F^i_j + q_j\,[-\Theta^i_j]^+, 0\} = \max\{F^i_j, F^i_j - q_j\,\Theta^i_j, 0\}$, the result follows by collecting the terms on the right side of the expression above and noting the definition of $V^i_t(\mathcal{R}^i(x_t))$ in (27). □

Lemma 4 *For all $t \in \mathcal{T}$, we have*

$$V_t(x_t) \le \psi^i_t(x_t) - \sum_{j\in\mathcal{J}} q_j\,\Lambda^i_j\, x_{jt} + \sum_{l\in\mathcal{L}\setminus\{i\}} \lambda^*_l\, c_l.$$

Proof of Lemma 4 We show the result by induction over the time periods. We let $\{w^*_j : j \in \mathcal{J}\}$ be the optimal solution to problem (1)-(4). We have

$$\begin{aligned} \Gamma(S(x_0)) = \sum_{j\in\mathcal{J}} \theta_j\, w^*_j &\ge \sum_{j\in\mathcal{J}} \theta_j\, w^*_j + \sum_{l\in\mathcal{L}\setminus\{i\}} \lambda^*_l \Big\{ \sum_{j\in\mathcal{J}} a_{lj}\,[S_j(x_{j0}) - w^*_j] - c_l \Big\} \\ &= \sum_{j\in\mathcal{J}} \Theta^i_j\, w^*_j + \sum_{j\in\mathcal{J}} \Lambda^i_j\, S_j(x_{j0}) - \sum_{l\in\mathcal{L}\setminus\{i\}} \lambda^*_l\, c_l \\ &\ge \phi^i(S(x_0)) + \sum_{j\in\mathcal{J}} \Lambda^i_j\, S_j(x_{j0}) - \sum_{l\in\mathcal{L}\setminus\{i\}} \lambda^*_l\, c_l, \end{aligned} \tag{47}$$

where the first inequality follows from the fact that the solution $\{w^*_j : j \in \mathcal{J}\}$ satisfies constraints (2) and $\lambda^*_l \ge 0$ for all $l \in \mathcal{L}\setminus\{i\}$, the second equality follows from (13) and the second inequality follows from the fact that $\{w^*_j : j \in \mathcal{J}\}$ is a feasible but not necessarily an optimal solution to problem (42)-(45). Taking expectations in the expression above, we obtain $V_0(x_0) = -\mathbb{E}\{\Gamma(S(x_0))\} \le -\mathbb{E}\{\phi^i(S(x_0))\} - \sum_{j\in\mathcal{J}} q_j\,\Lambda^i_j\, x_{j0} + \sum_{l\in\mathcal{L}\setminus\{i\}} \lambda^*_l\, c_l = \psi^i_0(x_0) - \sum_{j\in\mathcal{J}} q_j\,\Lambda^i_j\, x_{j0} + \sum_{l\in\mathcal{L}\setminus\{i\}} \lambda^*_l\, c_l$ and the result holds for the last time period. Assuming that the result holds for time period $t-1$, the induction assumption immediately implies that

$$\max\{f_j + V_{t-1}(x_t + e_j), V_{t-1}(x_t)\} \leq \max\{f_j + \psi^i_{t-1}(x_t + e_j) - q_j\,\Lambda^i_j, \psi^i_{t-1}(x_t)\} \\ - \sum_{j \in \mathcal{J}} q_j\,\Lambda^i_j\,x_{jt} + \sum_{l \in \mathcal{L}\setminus\{i\}} \lambda^*_l\,c_l.$$

Recalling that $F^i_j = f_j - q_j\,\Lambda^i_j$, one can combine the inequality above with (5) and (41) to obtain the result for time period t. □

We are now ready to finalize the proof of Proposition 2. Lemmas 3 and Lemma 4 imply that

$$V_t(x_t) \leq V^i_t(\mathcal{R}^i(x_t)) + \sum_{j \in \mathcal{J}\setminus\mathcal{J}^i} q_j\,[-\Theta^i_j]^+\,x_{jt} - \sum_{j \in \mathcal{J}} q_j\,\Lambda^i_j\,x_{jt} \\ + \sum_{j \in \mathcal{J}\setminus\mathcal{J}^i} \sum_{s=1}^{t} p_{js} \max\{F^i_j, F^i_j - q_j\,\Theta^i_j, 0\} + \sum_{l \in \mathcal{L}\setminus\{i\}} \lambda^*_l\,c_l.$$

The result follows by noting that the sum of the second and third terms on the right side of the expression above can be written as

$$\begin{aligned} \sum_{j \in \mathcal{J}\setminus\mathcal{J}^i} q_j\,[-\Theta^i_j]^+\,x_{jt} - \sum_{j \in \mathcal{J}} q_j\,\Lambda^i_j\,x_{jt} &= \sum_{j \in \mathcal{J}\setminus\mathcal{J}^i} q_j\,\max\{-\theta_j + \Lambda^i_j, 0\}\,x_{jt} - \sum_{j \in \mathcal{J}\setminus\mathcal{J}^i} q_j\,\Lambda^i_j\,x_{jt} - \sum_{j \in \mathcal{J}^i} q_j\,\Lambda^i_j\,x_{jt} \\ &= - \sum_{j \in \mathcal{J}\setminus\mathcal{J}^i} q_j\,\min\{\theta_j, \Lambda^i_j\}\,x_{jt} - \sum_{j \in \mathcal{J}^i} q_j\,\Lambda^i_j\,x_{jt} \\ &= - \sum_{j \in \mathcal{J}\setminus\mathcal{J}^i} \min\Big\{q_j\,\theta_j, q_j \sum_{l \in \mathcal{L}} a_{lj}\,\lambda^*_l\Big\}\,x_{jt} - \sum_{j \in \mathcal{J}^i} q_j\,\Lambda^i_j\,x_{jt}, \end{aligned}$$

where the last equality follows from the fact that $\Lambda^i_j = \sum_{l \in \mathcal{L}\setminus\{i\}} a_{lj}\,\lambda^*_l$ and $a_{ij} = 0$ whenever $j \in \mathcal{J}\setminus\mathcal{J}^i$.

C Appendix: Description of the Data Files

The data files that we use in our computational experiments are provided as an online supplement. The goal of this section is to describe the format of the data files. All of our data files are labeled as `rm_A_B_C.C_D.D_E.E_F.F.txt`, where `A` corresponds to the number of spokes in the airline network, `B` corresponds to the fare difference between a high fare and its corresponding low fare itinerary, `C.C` and `D.D` correspond to the parameters that we use to compute the penalty cost, `E.E` corresponds to the probability that a reservation shows up at the departure time and `F.F` corresponds to the ratio of the total expected demand to the total expected capacity. In other words, following the notation in Section 5.1, `A`, `B`, (`C.C`,`D.D`), `EE` and `F.F` respectively correspond to N, κ, (δ, σ), q and ρ.

In all of our data sets, we assume that we serve N spokes out of a single hub. Location 0 corresponds to the hub and locations $\{1, \ldots, N\}$ correspond to the spokes. The itineraries that connect the hub to a spoke or a spoke to the hub include one flight leg. The itineraries that connect two spokes include two flight legs, one from the origin spoke to the hub and one from the hub to the destination spoke.

Table 5 shows the organization of the data file for a test problem with $\tau = 3$ and $N = 2$. The character "#" indicates a comment line and such lines are skipped. The entries in the five portions of the data file have the following interpretations. The first portion of the data file shows the number of time periods in the planning horizon. The second portion of the data file shows the flight legs in the airline network. The first line in this portion shows the number of flight legs. After this first line, each line corresponds to one flight leg and shows the origin location, destination location and capacity of the flight leg. The third portion of the data file shows the itineraries. The first line in this portion shows the number of itineraries. After this first line, each line corresponds to one itinerary and shows the origin location, destination location, fare level, revenue and penalty cost for the itinerary. Fare level 0 indicates a low fare itinerary and fare level 1 indicates a high fare itinerary. We emphasize that the itineraries that connect two spokes include two flight legs, one from the origin spoke to the hub and one from the hub to the destination spoke. The fourth portion of the data file shows the arrival probabilities for the requests for different itineraries. Each line in this portion corresponds to a time period in the planning horizon. Each line first shows an itinerary indicated by the triplet [origin location, destination location, fare level], followed by the probability that we observe a request for this itinerary. For example, the probability that we observe a request for the low fare itinerary from location 2 to 1 at the first time period is 0.2. Since we may not observe any itinerary arrivals at a particular time period, the probabilities in a particular line do not necessarily add up to one. The fifth portion of the data file shows the show up probabilities. Each line in this portion corresponds to one itinerary. Each line first shows an itinerary indicated by the triplet [origin location, destination location, fare level], followed by the probability that a reservation for this itinerary shows up at the departure time.

```
# beginning of data file
# portion 1
# number of time periods in decision horizon
3

# portion 2
# list of flights [in format origin location, destination location, capacity]
# first line is number of flights
4
1 0 16
2 0 21
0 1 12
0 2 20

# portion 3
# list of itineraries [in format origin location, destination location, fare level, revenue, penalty cost]
# first line is number of itineraries
7
0 1 0 24.0 48.0
0 1 1 192.0 384.0
0 2 0 34.0 68.0
1 0 0 192.0 384.0
1 2 0 53.0 106.0
2 1 0 53.0 106.0
2 1 1 212.0 442.0

# portion 4
# list of request arrival probabilities [in format itinerary, probability]
# first entry in each line indicates time period
0  [0 1 0] 0.1    [0 1 1] 0.1    [0 2 0] 0.1    [1 0 0] 0.1    [1 2 0] 0.1    [2 1 0] 0.2    [2 1 1] 0.1
1  [0 1 0] 0.1    [0 1 1] 0.1    [0 2 0] 0.1    [1 0 0] 0.1    [1 2 0] 0.1    [2 1 0] 0.1    [2 1 1] 0.1
2  [0 1 0] 0.1    [0 1 1] 0.1    [0 2 0] 0.1    [1 0 0] 0.1    [1 2 0] 0.1    [2 1 0] 0.1    [2 1 1] 0.1

# portion 5
# list of show up probabilities [in format itinerary, probability]
[0 1 0] 0.9
[0 1 1] 0.9
[0 2 0] 0.9
[1 0 0] 0.9
[1 2 0] 0.9
[2 1 0] 0.9
[2 1 1] 0.9

# end of data file
```

Table 5: Organization of the data file for a test problem with $\tau = 3$ and $N = 2$.