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

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

November 22, 2007

Abstract

We propose a new method to compute bid prices in network revenue management problems. The novel aspect of our method is that it explicitly considers the temporal dynamics of the arrivals of the itinerary requests and generates bid prices that depend on the remaining leg capacities. Our method is based on relaxing certain constraints that link the decisions for different flight legs by associating Lagrange multipliers with them. In this case, the network revenue management problem decomposes by the flight legs and we can concentrate on one flight leg at a time. When compared with the so-called deterministic linear program, we show that our method provides a tighter upper bound on the optimal objective value of the network revenue management problem. Computational experiments indicate that the bid prices obtained by our method perform significantly better than the ones obtained by standard benchmark methods.

The idea of bid prices forms a powerful tool for solving network revenue management problems. This idea associates a bid price with each flight leg that captures the opportunity cost of a unit of capacity. An itinerary request is accepted only if there is enough capacity and the revenue from the itinerary request exceeds the sum of the bid prices associated with the flight legs that are in the requested itinerary; see Williamson (1992) and Talluri and van Ryzin (1998). One of the traditional approaches in the literature for computing bid prices involves solving a deterministic linear program. However, this linear program tends to be somewhat crude in the sense that it only uses the expected numbers of the itinerary requests that are to arrive until the time of departure and does not incorporate the probability distributions or temporal dynamics of the arrivals of the itinerary requests.

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

Our work builds on previous research. Hawkins (2003) and Adelman and Mersereau (2007) develop a Lagrangian relaxation method for what they call weakly coupled dynamic programs. In these dynamic programs, the evolutions of the different components of the state variable are affected by different types of decisions and these different types of decisions interact through a set of linking constraints. The authors propose relaxing the linking constraints by associating Lagrange multipliers with them. In this paper, we show that the network revenue management problem can be viewed as a weakly coupled dynamic program. In addition, the Lagrangian relaxation method in Hawkins (2003) and Adelman and Mersereau (2007) runs into computational difficulties when applied to the network revenue management problem. Specifically, this method requires finding a good set of Lagrange multipliers by minimizing the so-called dual function and the dual function may involve thousands of dimensions. We show that it is indeed possible to minimize the dual function efficiently by using standard subgradient optimization.

Network revenue management is an active area of research. The idea of bid prices dates back to Simpson (1989) and Williamson (1992), where the authors use the deterministic linear program mentioned above to compute bid prices. Talluri and van Ryzin (1998) give a careful analysis of the policies that are based on bid prices and point out that the idea of bid prices is equivalent to using linear value function approximations in the dynamic programming formulation of the network revenue management problem. Talluri and van Ryzin (1999) propose a randomized version of the deterministic linear program that uses actual samples of the itinerary requests that are to arrive until the time of departure. Their goal is to remedy the fact that the deterministic linear program only uses the expected numbers of the itinerary requests. Phillips (2005) describes a sequential estimation procedure to compute bid prices. This procedure builds on the popular expected marginal seat revenue heuristic of Belobaba (1987) and addresses the probabilistic nature of the itinerary requests. Bertsimas and Popescu (2003) propose a method that captures the total opportunity cost of the leg capacities consumed by an itinerary request more accurately. Their method essentially uses nonseparable and concave value function approximations. Adelman (2007) computes bid prices by using the linear programming representation of the dynamic programming formulation of the network revenue management problem. His approach explicitly considers the temporal dynamics of the arrivals of the itinerary requests, but does not generate bid prices that depend on the remaining leg capacities. Computational experiments indicate that the bid prices obtained by the methods proposed by Talluri and van Ryzin (1999), Bertsimas and Popescu (2003) and Adelman (2007) tend to perform better than the ones obtained by the deterministic linear program. Finally, other methods, besides bid prices, have been proposed for solving network revenue management problems. We do not go into the details of these methods and refer the reader to Talluri and van Ryzin (2004) for a comprehensive coverage of the network revenue management field.

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

The rest of the paper is organized as follows. Section 1 formulates the network revenue management problem as a dynamic program. Section 2 describes the Lagrangian relaxation idea and shows that our method provides an upper bound on the optimal objective value of the network revenue management problem. Section 3 establishes that our method obtains bid prices that depend on the remaining leg capacities. Section 4 shows that our method can efficiently find a good set of Lagrange multipliers by using standard subgradient optimization. Section 5 contrasts our method with the deterministic linear program. Section 6 presents computational experiments.

1 Problem Formulation

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

The problem takes place over the finite horizon $\mathcal{T} = \{1, \ldots, \tau\}$ and all flight legs depart at time period $\tau+1$. The set of flight legs is $\mathcal{L}$ and the set of itineraries is $\mathcal{J}$. If we accept a request for itinerary j, then we generate a revenue of f_j and consume a_{ij} units of capacity on flight leg i. If flight leg i is not in itinerary j, then we naturally have $a_{ij} = 0$. The initial capacity on flight leg i is c_{i1}. The probability that a request for itinerary j arrives into the system at time period t is p_{jt}. For notational brevity, we assume that $\sum_{j \in \mathcal{J}} p_{jt} = 1$ for all $t \in \mathcal{T}$. If there is a positive probability of having no itinerary requests at time period t, then we can handle this by defining a dummy itinerary ψ with $a_{i\psi} = 0$ for all $i \in \mathcal{L}$, $f_\psi = 0$ and $p_{\psi t} = 1 - \sum_{j \in \mathcal{J}} p_{jt}$. We assume that the itinerary requests at different time periods are independent of each other. Throughout the paper, we do not differentiate between column and row vectors. We use $|\mathcal{A}|$ to denote the cardinality of set $\mathcal{A}$.

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

$$\mathcal{U}(x_t) = \big\{ u_t \in \{0,1\}^{|\mathcal{J}|} : a_{ij}\, u_{jt} \le x_{it} \quad \forall\, i \in \mathcal{L},\ j \in \mathcal{J} \big\}.$$

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

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

where e_i is the $|\mathcal{L}|$-dimensional unit vector with a 1 in the element corresponding to $i \in \mathcal{L}$. Given the state variable x_t, it is easy to show that the optimal decisions at time period t are given by $u_t^*(x_t) = \{u_{jt}^*(x_t) : j \in \mathcal{J}\}$, where

$$u_{jt}^*(x_t) = \begin{cases} 1 & \text{if } f_j + V_{t+1}(x_t - \sum_{i \in \mathcal{L}} a_{ij}\, e_i) \ge V_{t+1}(x_t) \text{ and } a_{ij} \le x_{it} \text{ for all } i \in \mathcal{L} \\ 0 & \text{otherwise;} \end{cases} \tag{2}$$

see Adelman (2007). The main difficulty in solving the optimality equation in (1) arises from the fact that we have to make an "all or nothing" decision. Specifically, we either accept the itinerary request, in which case the capacities on all relevant flight legs are consumed, or we reject the itinerary request, in which case the leg capacities do not change. Our solution method is based on relaxing this requirement. In other words, when an itinerary request arrives, we allow ourselves to accept or reject the individual flight legs. We make this idea precise in the following section.

2 Lagrangian Relaxation

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

We begin by introducing some new notation. We augment $\mathcal{L}$ by a fictitious flight leg ϕ with infinite capacity. We extend the decisions at time period t as $y_t = \{y_{ijt} : i \in \mathcal{L} \cup \{\phi\},\ j \in \mathcal{J}\}$, where y_{ijt} takes value 1 if we accept flight leg i when a request for itinerary j arrives at time period t, and takes value 0 otherwise. In this case, it is easy to see that the optimality equation

$$V_t(x_t) = \max \quad \sum_{j \in \mathcal{J}} p_{jt} \Big\{ f_j\, y_{\phi jt} + V_{t+1}(x_t - \sum_{i \in \mathcal{L}} y_{ijt}\, a_{ij}\, e_i) \Big\} \tag{3}$$

$$\text{subject to} \quad a_{ij}\, y_{ijt} \le x_{it} \qquad \forall\, i \in \mathcal{L},\ j \in \mathcal{J} \tag{4}$$

$$y_{ijt} - y_{\phi jt} = 0 \qquad \forall\, i \in \mathcal{L},\ j \in \mathcal{J} \tag{5}$$

$$y_{ijt} \in \{0, 1\} \qquad \forall\, i \in \mathcal{L},\ j \in \mathcal{J} \tag{6}$$

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

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

$$V_t^\lambda(x_t) = \max \quad \sum_{j \in \mathcal{J}} p_{jt} \Big\{ \big[f_j - \sum_{i \in \mathcal{L}} \lambda_{ijt}\big]\, y_{\phi jt} + \sum_{i \in \mathcal{L}} \lambda_{ijt}\, y_{ijt} + V_{t+1}^\lambda(x_t - \sum_{i \in \mathcal{L}} y_{ijt}\, a_{ij}\, e_i) \Big\} \tag{7}$$

$$\text{subject to} \quad (4), (6) \tag{8}$$

$$y_{\phi jt} \in \{0, 1\} \qquad \forall\, j \in \mathcal{J}, \tag{9}$$

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

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

$$\mathcal{Y}_{it}(x_{it}) = \big\{ y_{it} \in \{0, 1\}^{|\mathcal{J}|} \, : \, a_{ij}\, y_{ijt} \le x_{it} \quad \forall\, j \in \mathcal{J} \big\},$$

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

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

$$\vartheta_{it}^\lambda(x_{it}) = \max_{y_{it} \in \mathcal{Y}_{it}(x_{it})} \left\{ \sum_{j \in \mathcal{J}} p_{jt} \Big\{ \lambda_{ijt}\, y_{ijt} + \vartheta_{i,t+1}^\lambda(x_{it} - a_{ij}\, y_{ijt}) \Big\} \right\} \tag{10}$$

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

$$V_t^\lambda(x_t) = \sum_{t'=t}^{\tau} \sum_{j \in \mathcal{J}} p_{jt'} \big[f_j - \sum_{i \in \mathcal{L}} \lambda_{ijt'}\big]^+ + \sum_{i \in \mathcal{L}} \vartheta_{it}^\lambda(x_{it}) \tag{11}$$

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

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

$$V_t^\lambda(x_t) = \max \quad \sum_{j \in \mathcal{J}} p_{jt} \Big\{ \big[f_j - \sum_{i \in \mathcal{L}} \lambda_{ijt}\big]\, y_{\phi jt} + \sum_{i \in \mathcal{L}} \lambda_{ijt}\, y_{ijt} + \sum_{i \in \mathcal{L}} \vartheta^\lambda_{i,t+1}(x_{it} - a_{ij}\, y_{ijt}) \Big\} + \sum_{t'=t+1}^{\tau} \sum_{j \in \mathcal{J}} p_{jt'} \big[f_j - \sum_{i \in \mathcal{L}} \lambda_{ijt'}\big]^+$$

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

$$y_{\phi jt} \in \{0, 1\} \qquad \forall\, j \in \mathcal{J}.$$

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

Therefore, we can efficiently solve the optimality equation in (7)-(9) by concentrating on one flight leg at a time. The terms on the right side of (11) can be interpreted as the value functions obtained from $|\mathcal{L}|+1$ revenue management problems, each of which involving only one flight leg. In particular, if we consider the revenue management problem that takes place over flight leg i without paying attention to the other flight legs and assume that we generate a revenue of λ_{ijt} whenever we accept a request for itinerary j at time period t, then $\{\vartheta^\lambda_{it}(x_{it}) : x_{it} \in \mathcal{C},\ t \in \mathcal{T}\}$ are the value functions associated with this revenue management problem. Similarly, if we consider the revenue management problem that takes place over the fictitious flight leg and assume that we generate a revenue of $f_j - \sum_{i \in \mathcal{L}} \lambda_{ijt}$ whenever we accept a request for itinerary j at time period t, then since the capacity on the fictitious flight leg is infinite, we accept an itinerary request whenever the revenue from the itinerary request is positive and $\{\sum_{t'=t}^{\tau} \sum_{j \in \mathcal{J}} p_{jt'}\, [f_j - \sum_{i \in \mathcal{L}} \lambda_{ijt'}]^+ : t \in \mathcal{T}\}$ are the value functions associated with this revenue management problem.

The following proposition shows that we obtain upper bounds on the value functions by solving the optimality equation in (7)-(9).

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

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

$$\begin{aligned} V_t(x_t) &= \sum_{j \in \mathcal{J}} p_{jt} \Big\{ f_j\, \hat{y}_{\phi jt} + V_{t+1}(x_t - \sum_{i \in \mathcal{L}} \hat{y}_{ijt}\, a_{ij}\, e_i) \Big\} \\ &= \sum_{j \in \mathcal{J}} p_{jt} \Big\{ \big[f_j - \sum_{i \in \mathcal{L}} \lambda_{ijt}\big]\, \hat{y}_{\phi jt} + \sum_{i \in \mathcal{L}} \lambda_{ijt}\, \hat{y}_{ijt} + V_{t+1}(x_t - \sum_{i \in \mathcal{L}} \hat{y}_{ijt}\, a_{ij}\, e_i) \Big\} \\ &\leq \sum_{j \in \mathcal{J}} p_{jt} \Big\{ \big[f_j - \sum_{i \in \mathcal{L}} \lambda_{ijt}\big]\, \hat{y}_{\phi jt} + \sum_{i \in \mathcal{L}} \lambda_{ijt}\, \hat{y}_{ijt} + V^\lambda_{t+1}(x_t - \sum_{i \in \mathcal{L}} \hat{y}_{ijt}\, a_{ij}\, e_i) \Big\} \leq V_t^\lambda(x_t), \end{aligned}$$

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

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

$$\min_\lambda \Big\{ V_1^\lambda(c_1) \Big\} = \min_\lambda \left\{ \sum_{t \in \mathcal{T}} \sum_{j \in \mathcal{J}} p_{jt} \big[f_j - \sum_{i \in \mathcal{L}} \lambda_{ijt} \big]^+ + \sum_{i \in \mathcal{L}} \vartheta_{i1}^\lambda(c_{i1}) \right\}, \tag{12}$$

where the equality follows from Proposition 1. The objective function of problem (12) is called the dual function. In Section 4, we show that the dual function is convex and problem (12) can be solved efficiently. However, before doing so, we take a quick detour to explain the structure of the policy obtained by approximating the value functions in (2) by using functions of the form (11).

3 Bid Price Structure of the Greedy Policy

Letting λ^* be an optimal solution to problem (12), our solution method approximates the value functions $\{V_t(x_t) : x_t \in \mathcal{C}^{|\mathcal{L}|},\ t \in \mathcal{T}\}$ in (2) by $\{V_t^{\lambda^*}(x_t) : x_t \in \mathcal{C}^{|\mathcal{L}|},\ t \in \mathcal{T}\}$. Specifically, given the state variable x_t, if we have

$$f_j + V_{t+1}^{\lambda^*}(x_t - \sum_{i \in \mathcal{L}} a_{ij}\, e_i) \geq V_{t+1}^{\lambda^*}(x_t) \tag{13}$$

and $a_{ij} \leq x_{it}$ for all $i \in \mathcal{L}$, then we accept a request for itinerary j at time period t. It is easy to see that this idea leads to bid prices that depend on the remaining leg capacities. Specifically, using Proposition 1, (13) can be written as

$$f_j \geq \sum_{i \in \mathcal{L}} \sum_{r=1}^{a_{ij}} \Big\{ \vartheta_{i,t+1}^{\lambda^*}(x_{it} + 1 - r) - \vartheta_{i,t+1}^{\lambda^*}(x_{it} - r) \Big\}. \tag{14}$$

In this case, we can view $\vartheta_{i,t+1}^{\lambda^*}(x_{it}) - \vartheta_{i,t+1}^{\lambda^*}(x_{it} - 1)$ as the bid price for the x_{it}-th unit of capacity on flight leg i. Similar to the idea of bid prices described in the introduction, if there is enough capacity and the revenue from an itinerary request exceeds the sum of the bid prices associated with the flight legs that are in the requested itinerary, then we accept the itinerary request. However, we emphasize that the bid price of a flight leg in (14) depends on the remaining leg capacity.

It is important to note that the separable structure of $\{V_t^{\lambda^*}(x_t) : x_t \in \mathcal{C}^{|\mathcal{L}|},\ t \in \mathcal{T}\}$ plays a major role in implementing the decision rule in (13) efficiently. In particular, if the separable structure did not exist, then implementing the decision rule in (13) would require storing $\{V_t^{\lambda^*}(x_t) : x_t \in \mathcal{C}^{|\mathcal{L}|},\ t \in \mathcal{T}\}$. This is equivalent to storing $|\mathcal{C}|^{|\mathcal{L}|}\, |\mathcal{T}|$ numbers. By using the separable structure, we can write (13) as (14) and implementing the decision rule in (14) requires storing $\{\vartheta_{it}^{\lambda^*}(x_{it}) : x_{it} \in \mathcal{C},\ i \in \mathcal{L},\ t \in \mathcal{T}\}$. This is equivalent to storing $|\mathcal{C}|\, |\mathcal{L}|\, |\mathcal{T}|$ numbers.

4 Minimizing the Dual Function

In this section, we show that $\vartheta^{\lambda}_{i1}(c_{i1})$ is a convex function of λ and its subgradients can be computed by solving the optimality equation in (10). This allows us to solve problem (12) by using standard subgradient optimization.

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

$$\vartheta^{\lambda}_{it}(x_{it}) = \sum_{j \in \mathcal{J}} p_{jt} \Big\{ \lambda_{ijt}\, y^{\lambda}_{ijt}(x_{it}) + \sum_{x_{i,t+1} \in \mathcal{C}} \mathbf{1}(x_{i,t+1} = x_{it} - a_{ij}\, y^{\lambda}_{ijt}(x_{it}))\; \vartheta^{\lambda}_{i,t+1}(x_{i,t+1}) \Big\}. \tag{15}$$

To write the expression above in matrix notation, we let Y^{λ}_{it} be the $|\mathcal{C}| \times |\mathcal{J}|$-dimensional matrix whose (x_{it}, j)-th element is $p_{jt}\, y^{\lambda}_{ijt}(x_{it})$ and Q^{λ}_{it} be the $|\mathcal{C}| \times |\mathcal{C}|$-dimensional matrix whose $(x_{it}, x_{i,t+1})$-th element is $\sum_{j \in \mathcal{J}} p_{jt}\, \mathbf{1}(x_{i,t+1} = x_{it} - a_{ij}\, y^{\lambda}_{ijt}(x_{it}))$. Letting ϑ^{λ}_{it} be the vector $\{\vartheta^{\lambda}_{it}(x_{it}) : x_{it} \in \mathcal{C}\}$ and λ_{it} be the vector $\{\lambda_{ijt} : j \in \mathcal{J}\}$, (15) can be written in matrix notation as

$$\vartheta^{\lambda}_{it} = Y^{\lambda}_{it}\, \lambda_{it} + Q^{\lambda}_{it}\, \vartheta^{\lambda}_{i,t+1}.$$

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

Proposition 3 *For any two sets of Lagrange multipliers λ and λ^o, we have*

$$\vartheta^{\lambda^o}_{it} \geq \vartheta^{\lambda}_{it} + Y^{\lambda}_{it}\big[\lambda^o_{it} - \lambda_{it}\big] + Q^{\lambda}_{it}\, Y^{\lambda}_{i,t+1}\big[\lambda^o_{i,t+1} - \lambda_{i,t+1}\big] + \ldots + Q^{\lambda}_{it}\, Q^{\lambda}_{i,t+1} \ldots Q^{\lambda}_{i,\tau-1}\, Y^{\lambda}_{i\tau}\big[\lambda^o_{i\tau} - \lambda_{i\tau}\big]$$

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

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

$$\vartheta^{\lambda}_{it}(x_{it}) = \sum_{j \in \mathcal{J}} p_{jt} \Big\{ \lambda_{ijt}\, y^{\lambda}_{ijt}(x_{it}) + \sum_{x_{i,t+1} \in \mathcal{C}} \mathbf{1}(x_{i,t+1} = x_{it} - a_{ij}\; y^{\lambda}_{ijt}(x_{it}))\; \vartheta^{\lambda}_{i,t+1}(x_{i,t+1}) \Big\}$$
$$\vartheta^{\lambda^o}_{it}(x_{it}) \geq \sum_{j \in \mathcal{J}} p_{jt} \Big\{ \lambda^o_{ijt}\, y^{\lambda}_{ijt}(x_{it}) + \sum_{x_{i,t+1} \in \mathcal{C}} \mathbf{1}(x_{i,t+1} = x_{it} - a_{ij}\; y^{\lambda}_{ijt}(x_{it}))\; \vartheta^{\lambda^o}_{i,t+1}(x_{i,t+1}) \Big\}.$$

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

$$\vartheta^{\lambda^o}_{it} - \vartheta^{\lambda}_{it} \geq Y^{\lambda}_{it}\,\big[\lambda^o_{it} - \lambda_{it}\big] + Q^{\lambda}_{it}\,\big[\vartheta^{\lambda^o}_{i,t+1} - \vartheta^{\lambda}_{i,t+1}\big]. \tag{16}$$

The result follows by using the induction assumption that

$$\vartheta^{\lambda^o}_{i,t+1} \geq \vartheta^{\lambda}_{i,t+1} + Y^{\lambda}_{i,t+1}\big[\lambda^o_{i,t+1} - \lambda_{i,t+1}\big] + Q^{\lambda}_{i,t+1}\, Y^{\lambda}_{i,t+2}\big[\lambda^o_{i,t+2} - \lambda_{i,t+2}\big]$$
$$+ \ldots + Q^{\lambda}_{i,t+1}\, Q^{\lambda}_{i,t+2} \ldots Q^{\lambda}_{i,\tau-1}\, Y^{\lambda}_{i\tau}\big[\lambda^o_{i\tau} - \lambda_{i\tau}\big]$$

in (16) and noting that the matrix Q^{λ}_{it} has positive elements. □

Letting $\Pi^{\lambda}_{it} = Q^{\lambda}_{i1}\, Q^{\lambda}_{i2} \ldots Q^{\lambda}_{i,t-1}\, Y^{\lambda}_{it}$ with $\Pi^{\lambda}_{i1} = Y^{\lambda}_{i1}$, we have

$$\vartheta^{\lambda^o}_{i1} \geq \vartheta^{\lambda}_{i1} + \Pi^{\lambda}_{i1}\left[\lambda^o_{i1} - \lambda_{i1}\right] + \Pi^{\lambda}_{i2}\left[\lambda^o_{i2} - \lambda_{i2}\right] + \ldots + \Pi^{\lambda}_{i\tau}\left[\lambda^o_{i\tau} - \lambda_{i\tau}\right]$$

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

$$\vartheta^{\lambda^o}_{i1}(c_{i1}) = \alpha_{i1}\, \vartheta^{\lambda^o}_{i1} \geq \alpha_{i1}\, \vartheta^{\lambda}_{i1} + \sum_{t \in \mathcal{T}} \alpha_{i1}\, \Pi^{\lambda}_{it}\left[\lambda^o_{it} - \lambda_{it}\right] = \vartheta^{\lambda}_{i1}(c_{i1}) + \sum_{t \in \mathcal{T}} \alpha_{i1}\, \Pi^{\lambda}_{it}\left[\lambda^o_{it} - \lambda_{it}\right].$$

Therefore, $\vartheta^{\lambda}_{i1}(c_{i1})$ has a subgradient and Theorem 3.2.6 in Bazaraa, Sherali and Shetty (1993) implies that $\vartheta^{\lambda}_{i1}(c_{i1})$ is a convex function of λ. The dual function, being a sum of convex functions of λ, is also a convex function of λ and we can use standard subgradient optimization to solve problem (12).

5 Relationship with the Deterministic Linear Program

An alternative solution method for the network revenue management problem described in Section 1 is to solve a deterministic linear program. Letting w_j be the number of requests for itinerary j that we plan to accept over the time periods $\{1, \ldots, \tau\}$, this linear program has the form

$$\max \quad \sum_{j \in \mathcal{J}} f_j\, w_j \tag{17}$$

$$\text{subject to} \quad \sum_{j \in \mathcal{J}} a_{ij}\, w_j \leq c_{i1} \qquad \forall\, i \in \mathcal{L} \tag{18}$$

$$w_j \leq \sum_{t \in \mathcal{T}} p_{jt} \qquad \forall\, j \in \mathcal{J} \tag{19}$$

$$w_j \geq 0 \qquad \forall\, j \in \mathcal{J}. \tag{20}$$

Constraints (18) ensure that the numbers of itinerary requests that we plan to accept do not violate the leg capacities, whereas constraints (19) ensure that we do not plan to accept more itinerary requests than the expected numbers of itinerary requests.

There are two main uses of problem (17)-(20). First, this problem can be used to decide whether we should accept or reject an itinerary request. In particular, letting $\{\mu^*_i : i \in \mathcal{L}\}$ be the optimal values of the dual variables associated with constraints (18), the idea is to use μ^*_i as the bid price associated with flight leg i. If there is enough capacity and the revenue from an itinerary request exceeds the sum of the bid prices associated with the flight legs that are in the requested itinerary, then we accept the itinerary request. Specifically, if we have

$$f_j \geq \sum_{i \in \mathcal{L}} a_{ij}\, \mu^*_i, \tag{21}$$

and $a_{ij} \leq x_{it}$ for all $i \in \mathcal{L}$, then we accept a request for itinerary j at time period t. Letting $\tilde{V}_t(x_t) = \sum_{i \in \mathcal{L}} \mu^*_i\, x_{it}$ for all $i \in \mathcal{L}$, $t \in \mathcal{T}$ and noting that $\tilde{V}_{t+1}(x_t) - \tilde{V}_{t+1}(x_t - \sum_{i \in \mathcal{L}} a_{ij}\, e_i) = \sum_{i \in \mathcal{L}} a_{ij}\, \mu^*_i$, it is easy to see that (21) is equivalent to approximating the value functions $\{V_t(x_t) : x_t \in \mathcal{C}^{|\mathcal{L}|},\ t \in \mathcal{T}\}$ in (2) by

$\{\tilde{V}_t(x_t) : x_t \in \mathcal{C}^{|\mathcal{L}|},\ t \in \mathcal{T}\}$. This approach is simple to implement and the computational experiments in Williamson (1992) indicate that it provides good solutions. Comparing (14) and (21), we emphasize that the bid prices obtained by solving problem (12) depend on the remaining leg capacities, whereas this is not the case for the bid prices obtained by solving problem (17)-(20).

Second, it is possible to show that the optimal objective value of problem (17)-(20) provides an upper bound on the maximum total expected revenue over the time periods $\{1, \ldots, \tau\}$; see Bertsimas and Popescu (2003). In other words, letting ζ^* be the optimal objective value of problem (17)-(20), we have $V_1(c_1) \leq \zeta^*$. This information can be useful when assessing the optimality gap of a suboptimal policy such as the one in (14) or (21).

In the remainder of this section, we show that

$$V_1(c_1) \leq \min_{\lambda} \Big\{ V_1^{\lambda}(c_1) \Big\} \leq \zeta^*. \tag{22}$$

Therefore, we can obtain a tighter upper bound on $V_1(c_1)$ by solving problem (12). Since the first inequality above follows from Proposition 2, we concentrate only on the second inequality.

Using the decision variables $\{z_{ij} : i \in \mathcal{L} \cup \{\phi\},\ j \in \mathcal{J}\}$, we write problem (17)-(20) as

$$\begin{aligned}
\max \quad & \sum_{j \in \mathcal{J}} f_j\, z_{\phi j} && && (23)\\
\text{subject to} \quad & \sum_{j \in \mathcal{J}} a_{ij}\, z_{ij} \leq c_{i1} && \forall\, i \in \mathcal{L} && (24)\\
& z_{\phi j} \leq \sum_{t \in \mathcal{T}} p_{jt} && \forall\, j \in \mathcal{J} && (25)\\
& z_{ij} - z_{\phi j} = 0 && \forall\, i \in \mathcal{L},\ j \in \mathcal{J} && (26)\\
& z_{\phi j},\ z_{ij} \geq 0 && \forall\, i \in \mathcal{L},\ j \in \mathcal{J}. && (27)
\end{aligned}$$

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

$$\begin{aligned}
\max \quad & \sum_{j \in \mathcal{J}} \big[f_j - \sum_{i \in \mathcal{L}} \beta^o_{ij} \big]\, z_{\phi j} + \sum_{i \in \mathcal{L}} \sum_{j \in \mathcal{J}} \beta^o_{ij}\, z_{ij}\\
\text{subject to} \quad & (24),\ (25),\ (27)
\end{aligned}$$

has the same optimal objective value as problem (23)-(27). Noting the upper and lower bounds on the decision variables $\{z_{\phi j} : j \in \mathcal{J}\}$, the problem above becomes

$$\begin{aligned}
\max \quad & \sum_{t \in \mathcal{T}} \sum_{j \in \mathcal{J}} p_{jt} \big[f_j - \sum_{i \in \mathcal{L}} \beta^o_{ij} \big]^+ + \sum_{i \in \mathcal{L}} \sum_{j \in \mathcal{J}} \beta^o_{ij}\, z_{ij} && && (28)\\
\text{subject to} \quad & \sum_{j \in \mathcal{J}} a_{ij}\, z_{ij} \leq c_{i1} && \forall\, i \in \mathcal{L} && (29)\\
& z_{ij} \geq 0 && \forall\, i \in \mathcal{L},\ j \in \mathcal{J}. && (30)
\end{aligned}$$

Letting $\{\mu_i : i \in \mathcal{L}\}$ be the dual variables associated with constraints (29), the dual of problem (28)-(30)

can be written as

$$
\begin{aligned}
\min \quad & \sum_{t\in\mathcal{T}}\sum_{j\in\mathcal{J}} p_{jt}\,\big[f_j-\sum_{i\in\mathcal{L}}\beta^o_{ij}\big]^+ + \sum_{i\in\mathcal{L}} c_{i1}\,\mu_i && (31)\\
\text{subject to} \quad & a_{ij}\,\mu_i \geq \beta^o_{ij} \qquad \forall\, i\in\mathcal{L},\ j\in\mathcal{J} && (32)\\
& \mu_i \geq 0 \qquad \forall\, i\in\mathcal{L}. && (33)
\end{aligned}
$$

Therefore, problem (31)-(33) has the same optimal objective value as problem (17)-(20). We are now ready to show that (22) holds.

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

Proof We let $\{\mu^*_i : i\in\mathcal{L}\}$ be an optimal solution to problem (31)-(33) and define the Lagrange multipliers $\lambda^o=\{\lambda^o_{ijt} : i\in\mathcal{L},\ j\in\mathcal{J},\ t\in\mathcal{T}\}$ as $\lambda^o_{ijt}=\beta^o_{ij}$ for all $i\in\mathcal{L}$, $j\in\mathcal{J}$, $t\in\mathcal{T}$. We begin by using induction over the time periods to show that $\vartheta^{\lambda^o}_{it}(x_{it})\leq \mu^*_i\,x_{it}$ for all $x_{it}\in\mathcal{C}$, $i\in\mathcal{L}$, $t\in\mathcal{T}$. For the last time period, we have

$$
\vartheta^{\lambda^o}_{i\tau}(x_{i\tau}) = \max_{y_{i\tau}\in\mathcal{Y}_{i\tau}(x_{i\tau})}\left\{\sum_{j\in\mathcal{J}} p_{j\tau}\,\beta^o_{ij}\,y_{ij\tau}\right\} \leq \max_{y_{i\tau}\in\mathcal{Y}_{i\tau}(x_{i\tau})}\left\{\sum_{j\in\mathcal{J}} p_{j\tau}\,\mu^*_i\,a_{ij}\,y_{ij\tau}\right\} \leq \sum_{j\in\mathcal{J}} p_{j\tau}\,\mu^*_i\,x_{i\tau} = \mu^*_i\,x_{i\tau},
$$

where the first inequality follows from (32) and the second inequality follows from the fact that $y_{i\tau}\in\mathcal{Y}_{i\tau}(x_{i\tau})$. Therefore, the result holds for the last time period. Assuming that the result holds for time period $t+1$, we have

$$
\begin{aligned}
\vartheta^{\lambda^o}_{it}(x_{it}) &= \max_{y_{it}\in\mathcal{Y}_{it}(x_{it})}\left\{\sum_{j\in\mathcal{J}} p_{jt}\left\{\beta^o_{ij}\,y_{ijt}+\vartheta^{\lambda^o}_{i,t+1}(x_{it}-a_{ij}\,y_{ijt})\right\}\right\}\\
&\leq \max_{y_{it}\in\mathcal{Y}_{it}(x_{it})}\left\{\sum_{j\in\mathcal{J}} p_{jt}\left\{\beta^o_{ij}\,y_{ijt}+\mu^*_i\left[x_{it}-a_{ij}\,y_{ijt}\right]\right\}\right\}\\
&= \max_{y_{it}\in\mathcal{Y}_{it}(x_{it})}\left\{\sum_{j\in\mathcal{J}} p_{jt}\left[\beta^o_{ij}-\mu^*_i\,a_{ij}\right]y_{ijt}\right\} + \mu^*_i\,x_{it} \leq \mu^*_i\,x_{it},
\end{aligned}
$$

where the first inequality follows from the induction assumption and the second inequality follows from (32). This establishes that $\vartheta^{\lambda^o}_{it}(x_{it})\leq \mu^*_i\,x_{it}$ for all $x_{it}\in\mathcal{C}$, $i\in\mathcal{L}$, $t\in\mathcal{T}$. In particular, we have $\vartheta^{\lambda^o}_{i1}(c_{i1})\leq\mu^*_i\,c_{i1}$ for all $i\in\mathcal{L}$, which implies that

$$
\begin{aligned}
\min_\lambda\left\{V_1^\lambda(c_1)\right\} \leq V_1^{\lambda^o}(c_1) &= \sum_{t\in\mathcal{T}}\sum_{j\in\mathcal{J}} p_{jt}\big[f_j-\sum_{i\in\mathcal{L}}\beta^o_{ij}\big]^+ + \sum_{i\in\mathcal{L}}\vartheta^{\lambda^o}_{i1}(c_{i1})\\
&\leq \sum_{t\in\mathcal{T}}\sum_{j\in\mathcal{J}} p_{jt}\big[f_j-\sum_{i\in\mathcal{L}}\beta^o_{ij}\big]^+ + \sum_{i\in\mathcal{L}}\mu^*_i\,c_{i1} = \zeta^*,
\end{aligned}
$$

where the first equality follows from Proposition 1 and the second equality follows by noting the objective function of problem (31)-(33). □

6 Computational Experiments

In this section, we compare the performance of our solution method with the performances of several benchmark strategies.

6.1 Benchmark Strategies

We compare the performances of the following six benchmark strategies.

Lagrangian relaxation (LR) This is the solution method that we develop in this paper, but our practical implementation refines the value function approximations five times over the decision horizon by solving problem (12) at time periods $\{1+k\,\tau/5 : k = 0, 1, \ldots, 4\}$. Specifically, given the state variable $x_{1+k\,\tau/5}$ at time period $1 + k\,\tau/5$, we solve the problem $\min_\lambda \big\{V^\lambda_{1+k\,\tau/5}(x_{1+k\,\tau/5})\big\}$ to obtain an optimal solution λ^{k*} and use $\{V_t^{\lambda^{k*}}(x_t) : x_t \in \mathcal{C}^{|\mathcal{L}|},\ t \in \mathcal{T}\}$ as approximations to the value functions until we solve problem (12) again.

We use standard subgradient optimization to solve the problem $\min_\lambda \big\{V^\lambda_{1+k\,\tau/5}(x_{1+k\,\tau/5})\big\}$; see Wolsey (1998). We initialize the step size parameter to $\sum_{j\in\mathcal{J}} f_j/|\mathcal{J}|$, and double the step size parameter after each iteration that results in a decrease in the objective function value and halve the step size parameter after each iteration that results in an increase in the objective function value. Although it does not guarantee convergence to an optimal solution, adjusting the step size parameter in this manner provides good solutions and stable performance.

Deterministic linear program (DLP) This is the solution method described in Section 5. Similar to LR, our practical implementation of DLP refines the bid prices by solving problem (17)-(20) five times over the decision horizon. Specifically, given the state variable $x_{1+k\,\tau/5}$ at time period $1 + k\,\tau/5$, we replace the right side of constraints (18) with $\{x_{i,1+k\,\tau/5} : i \in \mathcal{L}\}$ and the right side of constraints (19) with $\{\sum_{t=1+k\,\tau/5}^{\tau} p_{jt} : j \in \mathcal{J}\}$, and solve problem (17)-(20). Letting $\{\mu_i^* : i \in \mathcal{L}\}$ be the optimal values of the dual variables associated with constraints (18), we use $\{\mu_i^* : i \in \mathcal{L}\}$ as bid prices until we solve problem (17)-(20) again; see Talluri and van Ryzin (2004).

Randomized linear program (RLP) For brevity of discussion, we describe RLP and the following three solution methods under the assumption that the bid prices are computed only once at the beginning of the decision horizon. We emphasize that our practical implementations of these solution methods refine the bid prices five times over the decision horizon by using approaches similar to those that we use for LR and DLP.

DLP uses only the expected numbers of itinerary requests and RLP tries to make up for this shortcoming by using actual samples. In particular, we let D_{jt} be the number of requests for itinerary j at time period t so that we have $\mathbb{P}\{D_{jt} = 0\} = 1 - p_{jt}$ and $\mathbb{P}\{D_{jt} = 1\} = p_{jt}$. We generate S independent samples of $D = \{D_{jt} : j \in \mathcal{J},\ t \in \mathcal{T}\}$, which we denote by $\hat{D}^s = \{\hat{D}^s_{jt} : j \in \mathcal{J},\ t \in \mathcal{T}\}$ for $s = 1, \ldots, S$. We replace the right side of constraints (19) with $\{\sum_{t\in\mathcal{T}} \hat{D}^s_{jt} : j \in \mathcal{J}\}$ and solve problem (17)-(20). Letting $L_1(c_1, \hat{D}^s)$ be the optimal objective value of this problem and $\{\mu_i^{s*} : i \in \mathcal{L}\}$ be the optimal values of the dual variables associated with constraints (18), we use $\{\sum_{s=1}^{S} \mu_i^{s*}/S : i \in \mathcal{L}\}$ as bid prices;

see Talluri and van Ryzin (1999). Furthermore, it is possible to show that $V_1(c_1) \leq \mathbb{E}\{L_1(c_1, D)\}$. Therefore, RLP also provides an upper bound on the maximum total expected revenue over the time periods $\{1, \ldots, \tau\}$, but computing the expectation $\mathbb{E}\{L_1(c_1, D)\}$ requires estimation through simulation. We use $S = 50$ in our computational experiments.

Finite differences on deterministic linear program (DFD) The idea behind DFD is to try to capture the total opportunity cost of the leg capacities consumed by an itinerary request more accurately. In particular, we let $L_1(c_1)$ be the optimal objective value of problem (17)-(20). We replace the right side of constraints (18) with $\{c_{i1} - a_{ij} : i \in \mathcal{L}\}$ and solve problem (17)-(20) to obtain the optimal objective value $L^-_{j1}(c_1)$. If we have $f_j \geq L_1(c_1) - L^-_{j1}(c_1)$ and $a_{ij} \leq x_{it}$ for all $i \in \mathcal{L}$, then we accept a request for itinerary j at time period t; see Bertsimas and Popescu (2003).

Finite differences on randomized linear program (RFD) RFD is a natural extension of RLP and DFD, but it did not appear in the literature previously. Similar to RLP, we generate S independent samples of $D = \{D_{jt} : j \in \mathcal{J},\ t \in \mathcal{T}\}$, which we denote by $\hat{D}^s = \{\hat{D}^s_{jt} : j \in \mathcal{J},\ t \in \mathcal{T}\}$ for $s = 1, \ldots, S$. We replace the right side of constraints (19) with $\{\sum_{t \in \mathcal{T}} \hat{D}^s_{jt} : j \in \mathcal{J}\}$ and solve problem (17)-(20) to obtain the optimal objective value $L_1(c_1, \hat{D}^s)$. We then replace the right side of constraints (18) with $\{c_{i1} - a_{ij} : i \in \mathcal{L}\}$ and the right side of constraints (19) with $\{\sum_{t \in \mathcal{T}} \hat{D}^s_{jt} : j \in \mathcal{J}\}$, and solve problem (17)-(20) to obtain the optimal objective value $L^-_{j1}(c_1, \hat{D}^s)$. If we have $f_j \geq \sum_{s=1}^S \big[L_1(c_1, \hat{D}^s) - L^-_{j1}(c_1, \hat{D}^s)\big] / S$ and $a_{ij} \leq x_{it}$ for all $i \in \mathcal{L}$, then we accept a request for itinerary j at time period t. We use $S = 50$ in our computational experiments.

Linear value function approximations (LV) It is well-known that $V_1(c_1)$ can be computed by solving the linear program

$$
\begin{aligned}
\min \quad & V_1(c_1) \\
\text{subject to} \quad & V_t(x_t) \geq \sum_{j \in \mathcal{J}} p_{jt} \Big\{ f_j\, u_{jt} + V_{t+1}(x_t - u_{jt} \sum_{i \in \mathcal{L}} a_{ij}\, e_i) \Big\} && \forall\, x_t \in \mathcal{C}^{|\mathcal{L}|},\ u_t \in \mathcal{U}(x_t),\ t \in \mathcal{T} \setminus \{\tau\} \\
& V_\tau(x_\tau) \geq \sum_{j \in \mathcal{J}} p_{j\tau}\, f_j\, u_{j\tau} && \forall\, x_\tau \in \mathcal{C}^{|\mathcal{L}|},\ u_\tau \in \mathcal{U}(x_\tau),
\end{aligned}
$$

where $\{V_t(x_t) : x_t \in \mathcal{C}^{|\mathcal{L}|},\ t \in \mathcal{T}\}$ are the decision variables. One approach to deal with the large number of decision variables in the problem above is to approximate the value functions by linear functions of the form $\bar{V}_t(x_t) = \theta_t + \sum_{i \in \mathcal{L}} v_{it}\, x_{it}$. To decide what values to choose for $\{\theta_t : t \in \mathcal{T}\}$ and $\{v_{it} : i \in \mathcal{L},\ t \in \mathcal{T}\}$, we replace $V_t(x_t)$ with $\theta_t + \sum_{i \in \mathcal{L}} v_{it}\, x_{it}$ to obtain the linear program

$$
\begin{aligned}
\min \quad & \theta_1 + \sum_{i \in \mathcal{L}} v_{i1}\, c_{i1} \\
\text{subject to} \quad & \theta_t + \sum_{i \in \mathcal{L}} v_{it}\, x_{it} \geq \sum_{j \in \mathcal{J}} p_{jt} \Big\{ f_j\, u_{jt} + \theta_{t+1} + \sum_{i \in \mathcal{L}} v_{i,t+1} \big[x_{it} - a_{ij}\, u_{jt}\big] \Big\} \\
& && \forall\, x_t \in \mathcal{C}^{|\mathcal{L}|},\ u_t \in \mathcal{U}(x_t),\ t \in \mathcal{T} \setminus \{\tau\} \\
& \theta_\tau + \sum_{i \in \mathcal{L}} v_{i\tau}\, x_{i\tau} \geq \sum_{j \in \mathcal{J}} p_{j\tau}\, f_j\, u_{j\tau} && \forall\, x_\tau \in \mathcal{C}^{|\mathcal{L}|},\ u_\tau \in \mathcal{U}(x_\tau),
\end{aligned}
$$

where $\{\theta_t : t \in \mathcal{T}\}$ and $\{v_{it} : i \in \mathcal{L},\ t \in \mathcal{T}\}$ are the decision variables. The number of decision variables in the problem above is manageable and we can deal with the large number of constraints by

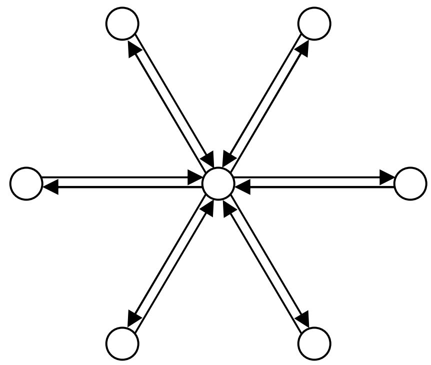

Figure 1: Structure of the network for the case where $N = 6$.

using constraint generation. Letting $\{\theta_t^* : t \in \mathcal{T}\}$ and $\{v_{it}^* : i \in \mathcal{L},\ t \in \mathcal{T}\}$ be an optimal solution to the problem above and $\bar{V}_t^*(x_t) = \theta_t^* + \sum_{i \in \mathcal{L}} v_{it}^* \, x_{it}$, we approximate the value functions $\{V_t(x_t) : x_t \in \mathcal{C}^{|\mathcal{L}|},\ t \in \mathcal{T}\}$ in (2) by $\{\bar{V}_t^*(x_t) : x_t \in \mathcal{C}^{|\mathcal{L}|},\ t \in \mathcal{T}\}$; see Adelman (2007). Furthermore, it is possible to show that $V_1(c_1) \leq \theta_1^* + \sum_{i \in \mathcal{L}} v_{i1}^* \, c_{i1}$. Therefore, LV also provides an upper bound on the maximum total expected revenue over the time periods $\{1, \ldots, \tau\}$.

6.2 Experimental Setup

We consider an airline network that serves N spokes out of a single hub. This is a key network structure that frequently arises in practice. Associated with each spoke, there are two flight legs, one of which is to the hub and the other one is from the hub. There is a high-fare and a low-fare itinerary that connects each origin-destination pair. Consequently, we have $2N$ flight legs and $2N(N + 1)$ itineraries, $4N$ of which involve one flight leg and $2N(N - 1)$ of which involve two flight legs. The revenue associated with each high-fare itinerary is κ times higher than the revenue associated with the corresponding low-fare itinerary. The probability of a request for each high-fare itinerary increases over time, whereas the probability of a request for each low-fare itinerary decreases over time. Since $\sum_{t \in \mathcal{T}} \sum_{j \in \mathcal{J}} p_{jt} \, a_{ij}$ is the total expected demand for the capacity on flight leg i, we measure the tightness of the leg capacities by

$$\alpha = \frac{\sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{L}} \sum_{j \in \mathcal{J}} p_{jt} \, a_{ij}}{\sum_{i \in \mathcal{L}} c_{i1}}.$$

Figure 1 shows the structure of the network for the case where $N = 6$.

We vary τ, N, α and κ to obtain different test problems and label our test problems by $(\tau, N, \alpha, \kappa) \in \{200, 600\} \times \{4, 5, 6, 8\} \times \{1.0, 1.2, 1.6\} \times \{4, 8\}$. To give a better feel for our experimental setup, Table 1 shows the average capacities per flight leg in our test problems. In this table, we note that two test problems that only differ in κ take place over the same network and use the same leg capacities. The online supplement provides the complete data for our test problems.

6.3 Computational Results

As mentioned in Sections 2, 5 and 6.1, LR, DLP, RLP and LV provide upper bounds on the maximum total expected revenue over the time periods $\{1, \ldots, \tau\}$. Tables 2 and 3 respectively show the upper

Problem $(\tau, N, \alpha, \kappa)$	Avg. cap. per leg
$(200, 4, 1.0, 4 \text{ or } 8)$	41
$(200, 4, 1.2, 4 \text{ or } 8)$	34
$(200, 4, 1.6, 4 \text{ or } 8)$	25
$(200, 5, 1.0, 4 \text{ or } 8)$	34
$(200, 5, 1.2, 4 \text{ or } 8)$	28
$(200, 5, 1.6, 4 \text{ or } 8)$	21
$(200, 6, 1.0, 4 \text{ or } 8)$	28
$(200, 6, 1.2, 4 \text{ or } 8)$	23
$(200, 6, 1.6, 4 \text{ or } 8)$	18
$(200, 8, 1.0, 4 \text{ or } 8)$	22
$(200, 8, 1.2, 4 \text{ or } 8)$	19
$(200, 8, 1.6, 4 \text{ or } 8)$	14

Problem $(\tau, N, \alpha, \kappa)$	Avg. cap per leg
$(600, 4, 1.0, 4 \text{ or } 8)$	61
$(600, 4, 1.2, 4 \text{ or } 8)$	51
$(600, 4, 1.6, 4 \text{ or } 8)$	38
$(600, 5, 1.0, 4 \text{ or } 8)$	51
$(600, 5, 1.2, 4 \text{ or } 8)$	42
$(600, 5, 1.6, 4 \text{ or } 8)$	32
$(600, 6, 1.0, 4 \text{ or } 8)$	34
$(600, 6, 1.2, 4 \text{ or } 8)$	28
$(600, 6, 1.6, 4 \text{ or } 8)$	21
$(600, 8, 1.0, 4 \text{ or } 8)$	27
$(600, 8, 1.2, 4 \text{ or } 8)$	22
$(600, 8, 1.6, 4 \text{ or } 8)$	17

Table 1: Average capacities per flight leg.

bounds obtained by different solution methods for the test problems with 200 and 600 time periods. The second, third, fourth and fifth columns in these tables respectively show the upper bounds obtained by LR, DLP, RLP and LV. Since it is not possible to compute the upper bound obtained by RLP explicitly, we provide a 95% confidence interval for $\mathbb{E}\{L_1(c_1, D)\}$ by using 10,000 samples. The sixth, seventh and eighth columns show the percent gaps between the upper bounds obtained by LR and the other three solution methods. The ninth column shows the CPU seconds required to solve problem (12) on a Pentium IV Desktop PC with 2.4 GHz CPU and 1 GB RAM running Windows XP. The tenth column shows the number of subgradient optimization iterations required to solve problem (12).

The results indicate that LR consistently provides the tightest upper bounds. On the average, the upper bounds obtained by LR are respectively 5.2%, 3.6% and 4.2% tighter than those obtained by DLP, RLP and LV. For the test problems with $\alpha = 1.0$, the upper bounds obtained by RLP are noticeably tighter than those obtained by LV, whereas for the test problems with $\alpha = 1.6$, the upper bounds obtained by LV are noticeably tighter than those obtained by RLP. Therefore, the tightness of the leg capacities seems to be an important factor affecting the quality of the upper bounds obtained by RLP and LV. It is not surprising that DLP consistently provides the loosest upper bounds. In particular, Proposition 4 in this paper, Section 4.1 in Talluri and van Ryzin (1998) and Theorem 1 in Adelman (2007) respectively show that the upper bounds obtained by LR, RLP and LV are provably tighter than the upper bound obtained by DLP.

Tables 4 and 5 respectively show the performances of the bid prices obtained by different solution methods for the test problems with 200 and 600 time periods. The second, third, fourth, fifth, sixth and seventh columns in these tables respectively show the total expected revenues obtained by LR, DLP, RLP, DFD, RFD and LV. We estimate these total expected revenues by simulating the performances of different solution methods under 100 demand trajectories. We use common random numbers when simulating the performances of different solution methods. The eighth, ninth, tenth, eleventh and twelfth columns show the percent gaps between the total expected revenues obtained by LR and the other five solution methods.

The results indicate that LR consistently provides the highest total expected revenues. The perfor-

Problem $(\tau, N, \alpha, \kappa)$	LR $(V_1^{\lambda^*}(c_1))$	DLP (ζ^*)	RLP $(\mathbb{E}\{L_1(c_1, D)\})$	LV $(\theta_1^* + \sum v_{i1}^* c_{i1})$	LR vs. DLP	LR vs. RLP	LR vs. LV	CPU	No. iter.
$(200, 4, 1.0, 4)$	20,439	21,531	$20,904 \mp 19$	21,348	5.3	2.3	4.4	103	855
$(200, 4, 1.0, 8)$	33,305	34,571	$33,947 \mp 41$	34,384	3.8	1.9	3.2	104	862
$(200, 4, 1.2, 4)$	18,938	19,882	$19,672 \mp 18$	19,663	5.0	3.9	3.8	106	1,066
$(200, 4, 1.2, 8)$	31,737	32,922	$32,715 \mp 40$	32,696	3.7	3.1	3.0	177	1,773
$(200, 4, 1.6, 4)$	16,600	17,530	$17,452 \mp 17$	17,303	5.6	5.1	4.2	93	1,269
$(200, 4, 1.6, 8)$	29,413	30,570	$30,494 \mp 40$	30,335	3.9	3.7	3.1	82	1,134
$(200, 5, 1.0, 4)$	21,298	22,144	$21,677 \mp 22$	22,016	4.0	1.8	3.4	161	1,051
$(200, 5, 1.0, 8)$	34,393	35,387	$34,903 \mp 45$	35,258	2.9	1.5	2.5	244	1,650
$(200, 5, 1.2, 4)$	20,184	21,263	$20,778 \mp 21$	21,108	5.3	2.9	4.6	78	626
$(200, 5, 1.2, 8)$	33,165	34,495	$33,989 \mp 45$	34,329	4.0	2.5	3.5	159	1,265
$(200, 5, 1.6, 4)$	17,704	18,870	$18,674 \mp 19$	18,565	6.6	5.5	4.9	46	525
$(200, 5, 1.6, 8)$	30,594	32,081	$31,875 \mp 43$	31,758	4.9	4.2	3.8	82	932
$(200, 6, 1.0, 4)$	21,128	22,300	$21,648 \mp 20$	22,116	5.5	2.5	4.7	231	1,353
$(200, 6, 1.0, 8)$	34,178	35,544	$34,890 \mp 43$	35,353	4.0	2.1	3.4	67	410
$(200, 6, 1.2, 4)$	19,649	20,932	$20,555 \mp 19$	20,649	6.5	4.6	5.1	106	778
$(200, 6, 1.2, 8)$	32,566	34,172	$33,792 \mp 42$	33,869	4.9	3.8	4.0	190	1,390
$(200, 6, 1.6, 4)$	17,304	18,592	$18,446 \mp 18$	18,565	7.4	6.6	7.3	107	1,021
$(200, 6, 1.6, 8)$	30,170	31,824	$31,679 \mp 41$	31,436	5.5	5.0	4.2	260	2,459
$(200, 8, 1.0, 4)$	18,975	20,052	$19,321 \mp 19$	19,870	5.7	1.8	4.7	45	200
$(200, 8, 1.0, 8)$	30,490	31,835	$31,086 \mp 40$	31,641	4.4	2.0	3.8	196	871
$(200, 8, 1.2, 4)$	17,472	18,952	$18,378 \mp 18$	18,598	8.5	5.2	6.4	245	1,300
$(200, 8, 1.2, 8)$	28,908	30,727	$30,142 \mp 40$	30,353	6.3	4.3	5.0	291	1,407
$(200, 8, 1.6, 4)$	15,295	16,833	$16,495 \mp 17$	16,378	10.1	7.8	7.1	205	1,370
$(200, 8, 1.6, 8)$	26,661	28,608	$28,255 \mp 39$	28,118	7.3	6.0	5.5	135	920

Table 2: Upper bounds on the maximum total expected revenue for the test problems with 200 time periods.

mance gaps between LR and the other five solution methods are statistically significant at 95% level for all of the test problems. For a majority of the test problems, RLP and RFD compete for the second and third places, and LV, DFD and DLP have respectively the fourth, fifth and sixth places. On the average, the total expected revenues obtained by LR are respectively 8.9%, 3.7%, 5.5%, 3.5% and 4.6% higher than those obtained by DLP, RLP, DFD, RFD and LV. It is especially surprising that the total expected revenues obtained by RLP and RFD can be noticeably higher than those obtained by LV. In particular, RLP and RFD use a simple randomization scheme on the deterministic linear program. The bid prices obtained by these two solution methods depend on the probability distributions of the total numbers of itinerary requests, but not on the order in which the itinerary requests arrive. On the other hand, the bid prices obtained by LV depend both on the probability distributions of the total numbers of itinerary requests and on the order in which the itinerary requests arrive.

To illustrate how different problem parameters affect the performance gaps, the five data series in Figure 2 plot the performance gaps between LR and the other five solution methods for all of the test problems. In this figure, blocks of six consecutive test problems in the horizontal axis share the same problem parameters other than the tightness of the leg capacities and the revenue difference between the high-fare and low-fare itineraries. The “triangular” patterns of especially the first, third and fifth data series indicate that the performance gaps between LR and DLP, LR and DFD, and LR and LV grow as the leg capacities get tighter and the revenue differences between the high-fare and low-fare itineraries get larger. For test problems with tight leg capacities and large revenue differences between

Problem $(\tau, N, \alpha, \kappa)$	LR $(V_1^{\lambda^*}(c_1))$	DLP (ζ^*)	RLP $(\mathbb{E}\{L_1(c_1, D)\})$	LV $(\theta_1^* + \sum v_{i1}^* c_{i1})$	LR vs. DLP	LR vs. RLP	LR vs. LV	CPU	No. iter.
$(600, 4, 1.0, 4)$	30,995	32,409	$31,579 \mp 34$	32,213	4.6	1.9	3.9	731	1,198
$(600, 4, 1.0, 8)$	50,444	52,086	$51,255 \mp 71$	51,876	3.3	1.6	2.8	844	1,371
$(600, 4, 1.2, 4)$	28,668	29,852	$29,642 \mp 30$	29,618	4.1	3.4	3.3	333	693
$(600, 4, 1.2, 8)$	48,054	49,529	$49,317 \mp 68$	49,279	3.1	2.6	2.6	985	2,012
$(600, 4, 1.6, 4)$	25,148	26,324	$26,253 \mp 29$	26,082	4.7	4.4	3.7	1,301	3,820
$(600, 4, 1.6, 8)$	44,555	46,001	$45,928 \mp 66$	45,742	3.2	3.1	2.7	494	1,459
$(600, 5, 1.0, 4)$	32,254	33,299	$32,723 \mp 38$	33,153	3.2	1.5	2.8	1,431	2,026
$(600, 5, 1.0, 8)$	52,071	53,285	$52,685 \mp 76$	53,134	2.3	1.2	2.0	1,211	1,681
$(600, 5, 1.2, 4)$	30,604	31,943	$31,404 \mp 34$	31,773	4.4	2.6	3.8	2,430	4,077
$(600, 5, 1.2, 8)$	50,282	51,904	$51,340 \mp 73$	51,717	3.2	2.1	2.9	669	1,160
$(600, 5, 1.6, 4)$	26,936	28,343	$28,183 \mp 30$	28,022	5.2	4.6	4.0	462	1,116
$(600, 5, 1.6, 8)$	46,497	48,283	$48,105 \mp 70$	47,939	3.8	3.5	3.1	297	718
$(600, 6, 1.0, 4)$	25,541	26,873	$26,130 \mp 34$	26,722	5.2	2.3	4.6	1,006	1,605
$(600, 6, 1.0, 8)$	41,412	42,865	$42,113 \mp 69$	42,703	3.5	1.7	3.1	1,031	1,638
$(600, 6, 1.2, 4)$	23,687	25,184	$24,756 \mp 30$	24,878	6.3	4.5	5.0	745	1,476
$(600, 6, 1.2, 8)$	39,307	41,166	$40,732 \mp 66$	40,834	4.7	3.6	3.9	812	1,623
$(600, 6, 1.6, 4)$	20,817	22,274	$22,132 \mp 28$	21,893	7.0	6.3	5.2	463	1,210
$(600, 6, 1.6, 8)$	36,391	38,252	$38,103 \mp 64$	37,842	5.1	4.7	4.0	1,231	3,179
$(600, 8, 1.0, 4)$	22,960	24,167	$23,375 \mp 31$	23,998	5.3	1.8	4.5	11	20
$(600, 8, 1.0, 8)$	36,933	38,395	$37,595 \mp 64$	38,217	4.0	1.8	3.5	10	20
$(600, 8, 1.2, 4)$	21,102	22,755	$22,150 \mp 28$	22,382	7.8	5.0	6.1	1,076	1,573
$(600, 8, 1.2, 8)$	34,931	36,976	$36,368 \mp 62$	36,580	5.9	4.1	4.7	857	1,232
$(600, 8, 1.6, 4)$	18,500	20,228	$19,890 \mp 26$	19,761	9.3	7.5	6.8	974	1,816
$(600, 8, 1.6, 8)$	32,247	34,449	$34,105 \mp 59$	33,942	6.8	5.8	5.3	717	1,347

Table 3: Upper bounds on the maximum total expected revenue for the test problems with 600 time periods.

the high-fare and low-fare itineraries, the cost of making an "incorrect" decision is quite high. For these test problems, LR provides especially good performance when compared with the other solution methods.

The performances of DLP and DFD can be sensitive to the number of times that we refine the bid prices; see Talluri and van Ryzin (2004). Tables 6 and 7 respectively show the performances of DLP and DFD for the test problems with 200 and 600 time periods when we refine the bid prices 20 times over the decision horizon. For comparison purpose, the second column in these tables shows the total expected revenues obtained by LR when we refine the value function approximations five times over the decision horizon. The third and fourth columns respectively show the total expected revenues obtained by DLP and DFD when we refine the bid prices 20 times over the decision horizon. The fifth and sixth columns show the percent gaps between the total expected revenues obtained by LR and the other two solution methods. Comparing the total expected revenues obtained by DLP and DFD in Tables 6 and 7 with those in Tables 4 and 5, we note that the performances of DLP and DFD improve noticeably when we refine the bid prices more frequently. Nevertheless, the total expected revenues obtained by LR are still significantly higher than those obtained by DLP and DFD.

To give a feel for how the computational effort for LR scales with problem size, Figure 3 shows the CPU seconds required for one subgradient optimization iteration as a function of the average capacity per flight leg. In this figure, we consider test problems with $\tau = 600$ and $N = 8$. The results indicate

Problem $(\tau, N, \alpha, \kappa)$	LR	DLP	RLP	DFD	RFD	LV	LR vs. DLP	LR vs. RLP	LR vs. DFD	LR vs. RFD	LR vs. LV
$(200, 4, 1.0, 4)$	20,018	19,367	19,634	19,573	19,576	19,572	3.3	1.9	2.2	2.2	2.2
$(200, 4, 1.0, 8)$	32,626	30,713	31,671	31,316	31,764	31,523	5.9	2.9	4.0	2.6	3.4
$(200, 4, 1.2, 4)$	18,374	17,082	17,643	17,631	17,742	17,829	7.0	4.0	4.0	3.4	3.0
$(200, 4, 1.2, 8)$	30,852	27,238	29,413	29,028	29,796	29,453	11.7	4.7	5.9	3.4	4.5
$(200, 4, 1.6, 4)$	15,981	14,251	15,444	15,101	15,413	15,148	10.8	3.4	5.5	3.6	5.2
$(200, 4, 1.6, 8)$	28,381	23,573	27,204	25,912	27,414	26,160	16.9	4.1	8.7	3.4	7.8
$(200, 5, 1.0, 4)$	21,181	20,143	20,708	20,457	20,679	20,742	4.9	2.2	3.4	2.4	2.1
$(200, 5, 1.0, 8)$	34,271	31,881	33,368	32,575	33,463	33,083	7.0	2.6	4.9	2.4	3.5
$(200, 5, 1.2, 4)$	19,818	18,619	19,253	19,127	19,292	19,315	6.1	2.9	3.5	2.7	2.5
$(200, 5, 1.2, 8)$	32,766	29,567	31,551	30,849	31,766	31,398	9.8	3.7	5.8	3.1	4.2
$(200, 5, 1.6, 4)$	17,318	15,432	16,592	16,420	16,708	16,561	10.9	4.2	5.2	3.5	4.4
$(200, 5, 1.6, 8)$	30,107	24,998	28,628	26,890	29,150	28,541	17.0	4.9	10.7	3.2	5.2
$(200, 6, 1.0, 4)$	20,709	19,789	20,195	20,015	20,195	20,167	4.4	2.5	3.3	2.5	2.6
$(200, 6, 1.0, 8)$	33,466	31,084	32,421	31,821	32,565	32,253	7.1	3.1	4.9	2.7	3.6
$(200, 6, 1.2, 4)$	19,133	18,063	18,451	18,414	18,501	18,578	5.6	3.6	3.8	3.3	2.9
$(200, 6, 1.2, 8)$	31,808	28,662	30,386	29,862	30,616	30,373	9.9	4.5	6.1	3.7	4.5
$(200, 6, 1.6, 4)$	16,769	15,250	16,045	15,896	16,115	16,082	9.1	4.3	5.2	3.9	4.1
$(200, 6, 1.6, 8)$	29,320	24,920	27,792	27,067	28,275	27,386	15.0	5.2	7.7	3.6	6.6
$(200, 8, 1.0, 4)$	18,217	17,245	17,650	17,536	17,703	17,583	5.3	3.1	3.7	2.8	3.5
$(200, 8, 1.0, 8)$	29,453	26,973	28,288	27,919	28,573	27,873	8.4	4.0	5.2	3.0	5.4
$(200, 8, 1.2, 4)$	16,941	15,615	16,036	16,132	16,291	16,092	7.8	5.3	4.8	3.8	5.0
$(200, 8, 1.2, 8)$	28,130	24,564	26,399	26,092	26,972	26,021	12.7	6.2	7.2	4.1	7.5
$(200, 8, 1.6, 4)$	14,720	13,335	13,919	13,970	14,131	13,732	9.4	5.4	5.1	4.0	6.7
$(200, 8, 1.6, 8)$	25,701	21,584	24,173	23,709	24,756	23,249	16.0	5.9	7.8	3.7	9.5

Table 4: Total expected revenues for the test problems with 200 time periods.

that the CPU seconds required for one subgradient optimization iteration scale slightly worse than linearly with the average capacity per flight leg.

7 Conclusions

In this paper, we developed a new method to compute bid prices in network revenue management problems. The novel aspect of our method is that it generates bid prices that depend on how much time is left until the time of departure and how much capacity is left on the flight legs. Our method naturally decomposes the network revenue management problem by the flight legs and allows us to compute bid prices by concentrating on one flight leg at a time. When compared with the linear programming-based methods for computing bid prices, our method requires more computational power, but it noticeably improves the upper bounds on the maximum total expected revenues and provides bid prices that obtain significantly higher total expected revenues.

Acknowledgements The author thanks the associate editor and two anonymous referees for their useful comments that especially strengthened the computational results. This work was supported in part by National Science Foundation grant DMI-0422133.

References

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

Adelman, D. and Mersereau, A. J. (2007), 'Relaxations of weakly coupled stochastic dynamic programs', *Operations Research* (to appear).

Problem $(\tau, N, \alpha, \kappa)$	LR	DLP	RLP	DFD	RFD	LV	LR vs. DLP	LR vs. RLP	LR vs. DFD	LR vs. RFD	LR vs. LV
$(600, 4, 1.0, 4)$	30,640	29,661	29,926	29,816	30,056	30,063	3.2	2.3	2.7	1.9	1.9
$(600, 4, 1.0, 8)$	49,862	47,106	48,426	47,608	48,818	48,952	5.5	2.9	4.5	2.1	1.8
$(600, 4, 1.2, 4)$	28,145	26,366	27,261	26,883	27,024	27,298	6.3	3.1	4.5	4.0	3.0
$(600, 4, 1.2, 8)$	47,162	42,258	45,602	43,955	45,351	44,606	10.4	3.3	6.8	3.8	5.4
$(600, 4, 1.6, 4)$	24,540	22,177	23,987	22,860	23,776	23,306	9.6	2.3	6.8	3.1	5.0
$(600, 4, 1.6, 8)$	43,547	37,019	42,589	38,779	42,141	40,183	15.0	2.2	11.0	3.2	7.7
$(600, 5, 1.0, 4)$	32,112	30,701	31,523	31,081	31,723	31,469	4.4	1.8	3.2	1.2	2.0
$(600, 5, 1.0, 8)$	51,875	48,576	50,661	49,434	49,995	50,786	6.4	2.3	4.7	3.6	2.1
$(600, 5, 1.2, 4)$	30,308	28,567	29,463	29,242	29,153	29,579	5.7	2.8	3.5	3.8	2.4
$(600, 5, 1.2, 8)$	49,899	45,518	48,206	46,721	47,857	47,950	8.8	3.4	6.4	4.1	3.9
$(600, 5, 1.6, 4)$	26,605	24,195	25,641	24,801	25,333	25,449	9.1	3.6	6.8	4.8	4.3
$(600, 5, 1.6, 8)$	46,070	39,623	44,456	41,665	43,887	42,873	14.0	3.5	9.6	4.7	6.9
$(600, 6, 1.0, 4)$	25,310	24,185	24,702	24,461	24,424	24,623	4.4	2.4	3.4	3.5	2.7
$(600, 6, 1.0, 8)$	40,849	38,068	39,596	38,648	39,225	39,293	6.8	3.1	5.4	4.0	3.8
$(600, 6, 1.2, 4)$	23,306	21,766	22,437	22,343	22,377	22,474	6.6	3.7	4.1	4.0	3.6
$(600, 6, 1.2, 8)$	38,704	34,533	37,008	35,898	37,051	36,312	10.8	4.4	7.2	4.3	6.2
$(600, 6, 1.6, 4)$	20,273	18,441	19,373	19,287	19,445	19,366	9.0	4.4	4.9	4.1	4.5
$(600, 6, 1.6, 8)$	35,631	30,370	33,599	32,005	34,119	32,860	14.8	5.7	10.2	4.2	7.8
$(600, 8, 1.0, 4)$	22,269	21,243	21,554	21,657	21,515	21,607	4.6	3.2	2.7	3.4	3.0
$(600, 8, 1.0, 8)$	36,046	33,274	34,604	34,274	34,665	34,104	7.7	4.0	4.9	3.8	5.4
$(600, 8, 1.2, 4)$	20,643	19,150	19,706	19,626	19,697	19,728	7.2	4.5	4.9	4.6	4.4
$(600, 8, 1.2, 8)$	34,277	30,237	32,369	31,992	32,654	31,772	11.8	5.6	6.7	4.7	7.3
$(600, 8, 1.6, 4)$	17,930	16,407	17,061	17,088	17,105	16,905	8.5	4.9	4.7	4.6	5.7
$(600, 8, 1.6, 8)$	31,317	26,815	29,609	28,932	29,825	28,469	14.4	5.5	7.6	4.8	9.1

Table 5: Total expected revenues for the test problems with 600 time periods.

Bazaraa, M. S., Sherali, H. D. and Shetty, C. M. (1993), *Nonlinear Programming: Theory and Algorithms*, second edn, John Wiley & Sons, Inc., New York.

Belobaba, P. P. (1987), Air Travel Demand and Airline Seat Inventory Control, PhD thesis, MIT, Cambridge, MA.

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

Hawkins, J. (2003), A Lagrangian Decomposition Approach to Weakly Coupled Dynamic Optimization Problems and its Applications, PhD thesis, Massachusetts Institute of Technology, Cambridge, MA.

Phillips, R. (2005), *Pricing and Revenue Optimization*, Stanford University Press, Stanford, CA.

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

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

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

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

Wolsey, L. A. (1998), *Integer Programming*, John Wiley & Sons, Inc., New York.

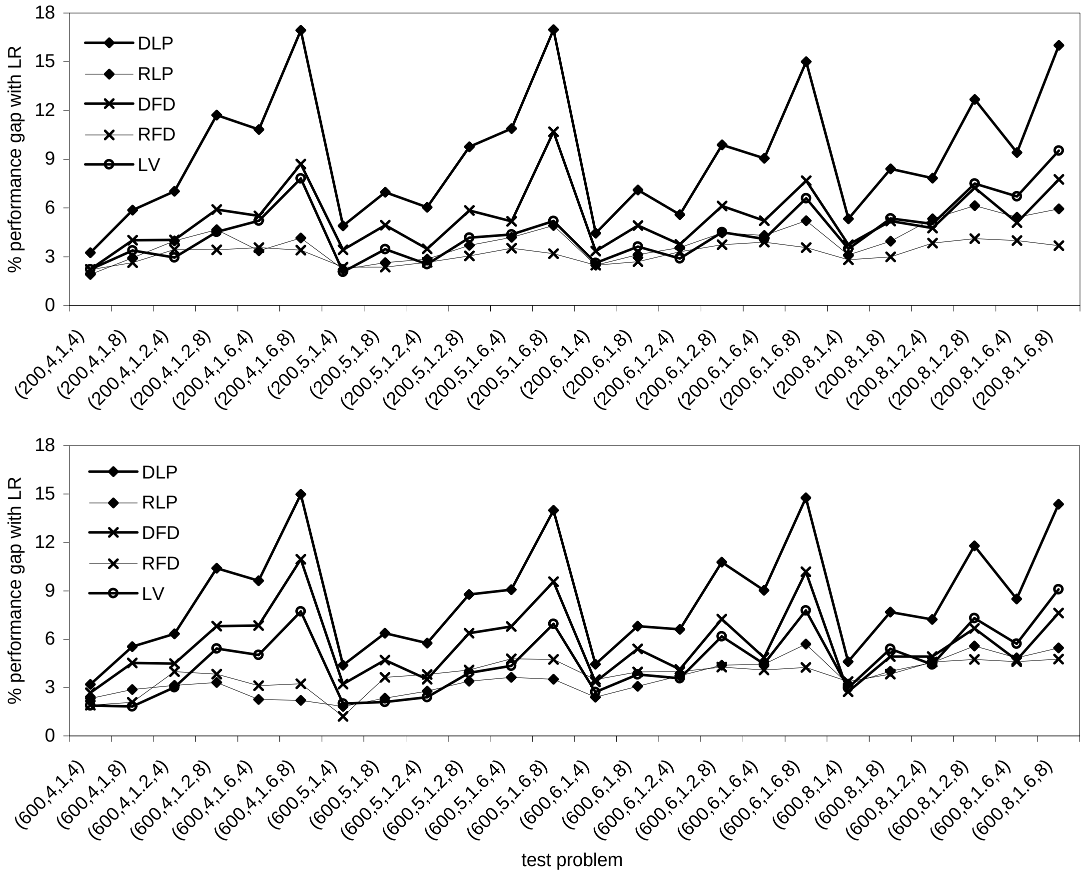

Figure 2: Performance gaps between LR and the other five solution methods.

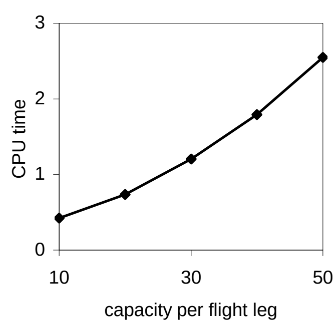

Figure 3: CPU seconds required for one subgradient optimization iteration as a function of the average capacity per flight leg.

Problem $(\tau, N, \alpha, \kappa)$	LR	DLP	DFD	LR vs. DLP	LR vs. DFD
$(200, 4, 1.0, 4)$	20,018	19,691	19,772	1.6	1.2
$(200, 4, 1.0, 8)$	32,626	31,453	31,823	3.6	2.5
$(200, 4, 1.2, 4)$	18,374	17,661	18,027	3.9	1.9
$(200, 4, 1.2, 8)$	30,852	28,566	29,583	7.4	4.1
$(200, 4, 1.6, 4)$	15,981	15,110	15,559	5.5	2.6
$(200, 4, 1.6, 8)$	28,381	25,581	26,901	9.9	5.2
$(200, 5, 1.0, 4)$	21,181	20,503	20,708	3.2	2.2
$(200, 5, 1.0, 8)$	34,271	32,597	33,134	4.9	3.3
$(200, 5, 1.2, 4)$	19,818	18,988	19,406	4.2	2.1
$(200, 5, 1.2, 8)$	32,766	30,417	31,396	7.2	4.2
$(200, 5, 1.6, 4)$	17,318	16,301	16,838	5.9	2.8
$(200, 5, 1.6, 8)$	30,107	26,997	28,236	10.3	6.2
$(200, 6, 1.0, 4)$	20,709	20,152	20,285	2.7	2.0
$(200, 6, 1.0, 8)$	33,466	31,886	32,315	4.7	3.4
$(200, 6, 1.2, 4)$	19,133	18,449	18,716	3.6	2.2
$(200, 6, 1.2, 8)$	31,808	29,542	30,470	7.1	4.2
$(200, 6, 1.6, 4)$	16,769	15,828	16,243	5.6	3.1
$(200, 6, 1.6, 8)$	29,320	26,305	27,755	10.3	5.3
$(200, 8, 1.0, 4)$	18,217	17,599	17,861	3.4	2.0
$(200, 8, 1.0, 8)$	29,453	27,731	28,390	5.8	3.6
$(200, 8, 1.2, 4)$	16,941	16,114	16,432	4.9	3.0
$(200, 8, 1.2, 8)$	28,130	25,625	26,684	8.9	5.1
$(200, 8, 1.6, 4)$	14,720	13,830	14,265	6.0	3.1
$(200, 8, 1.6, 8)$	25,701	22,780	24,178	11.4	5.9

Table 6: Total expected revenues obtained by DLP and DFD for the test problems with 200 time periods when we refine the bid prices 20 times over the decision horizon.

Problem $(\tau, N, \alpha, \kappa)$	LR	DLP	DFD	LR vs. DLP	LR vs. DFD
$(600, 4, 1.0, 4)$	30,640	30,131	30,258	1.7	1.2
$(600, 4, 1.0, 8)$	49,862	48,239	48,698	3.3	2.3
$(600, 4, 1.2, 4)$	28,145	27,254	27,506	3.2	2.3
$(600, 4, 1.2, 8)$	47,162	44,340	45,504	6.0	3.5
$(600, 4, 1.6, 4)$	24,540	23,394	23,758	4.7	3.2
$(600, 4, 1.6, 8)$	43,547	39,862	40,963	8.5	5.9
$(600, 5, 1.0, 4)$	32,112	31,078	31,328	3.2	2.4
$(600, 5, 1.0, 8)$	51,875	49,459	50,075	4.7	3.5
$(600, 5, 1.2, 4)$	30,308	29,100	29,425	4.0	2.9
$(600, 5, 1.2, 8)$	49,899	46,788	47,668	6.2	4.5
$(600, 5, 1.6, 4)$	26,605	25,073	25,560	5.8	3.9
$(600, 5, 1.6, 8)$	46,070	41,711	43,145	9.5	6.3
$(600, 6, 1.0, 4)$	25,310	24,644	24,880	2.6	1.7
$(600, 6, 1.0, 8)$	40,849	39,061	39,570	4.4	3.1
$(600, 6, 1.2, 4)$	23,306	22,356	22,814	4.1	2.1
$(600, 6, 1.2, 8)$	38,704	35,790	36,841	7.5	4.8
$(600, 6, 1.6, 4)$	20,273	19,150	19,704	5.5	2.8
$(600, 6, 1.6, 8)$	35,631	31,884	33,295	10.5	6.6
$(600, 8, 1.0, 4)$	22,269	21,621	21,907	2.9	1.6
$(600, 8, 1.0, 8)$	36,046	34,075	34,877	5.5	3.2
$(600, 8, 1.2, 4)$	20,643	19,601	20,140	5.0	2.4
$(600, 8, 1.2, 8)$	34,277	31,347	32,650	8.5	4.7
$(600, 8, 1.6, 4)$	17,930	16,939	17,423	5.5	2.8
$(600, 8, 1.6, 8)$	31,317	28,086	29,572	10.3	5.6

Table 7: Total expected revenues obtained by DLP and DFD for the test problems with 600 time periods when we refine the bid prices 20 times over the decision horizon.