Transportation Science

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

A Network Airline Revenue Management Framework Based on Decomposition by Origins and Destinations

Ş. İlker Birbil, J. B. G. Frenk, Joaquim A. S. Gromicho, Shuzhong Zhang

TRANSPORTATION SCIENCE
Vol. 48, No. 3, August 2014, pp. 313–333
ISSN 0041-1655 (print) | ISSN 1526-5447 (online)
http://dx.doi.org/10.1287/trsc.2013.0469

A Network Airline Revenue Management Framework Based on Decomposition by Origins and Destinations

Ş. İlker Birbil, J. B. G. Frenk
Faculty of Engineering and Natural Sciences, Sabancı University, Orhanlı-Tuzla, 34956 Istanbul, Turkey
{sibirbil@sabanciuniv.edu, frenk@sabanciuniv.edu}

Joaquim A. S. Gromicho
VU University, 1081 HV Amsterdam, The Netherlands; and ORTEC, 2719 EA Zoetermeer, The Netherlands,
j.a.dossantos.gromicho@vu.nl

Shuzhong Zhang
Department of Industrial and Systems Engineering, University of Minnesota, Minneapolis, Minnesota 55455,
zhangs@umn.edu

We propose a framework for solving airline revenue management problems on large networks, where the main concern is to allocate the flight leg capacities to customer requests under fixed class fares. This framework is based on a mathematical programming model that decomposes the network into origin-destination pairs so that each pair can be treated as a single flight-leg problem. We first discuss that the proposed framework is quite generic in the sense that not only several well-known models from the literature fit into this framework, but also many single flight-leg models can be easily extended to a network setting through the prescribed construction. Then, we analyze the structure of the overall mathematical programming model and establish its relationship with other models frequently used in practice. The application of the proposed framework is illustrated through two examples based on static and dynamic single-leg models, respectively. These illustrative examples are then benchmarked against several existing methods on a set of real-life network problems.

Keywords: airline network revenue management; generic framework; decomposition; computational study
History: Received: March 2012; revisions received: August 2012, November 2012; accepted: February 2013. Published online in *Articles in Advance* July 30, 2013.

1. Introduction

Historically, the early adopters of computer reservations systems were mainly from the airline industry. Having these systems at hand, the airline executives had the chance to investigate the efficient ways for managing their limited seat inventories. Particularly after the deregulations in the industry, the revenue management techniques have become indispensable for airline capacity control. Naturally, the increase in the airline revenues has not been left unnoticed by other service industries including tourism, transportation, entertainment, etc. This considerable attention in practice has also motivated the researchers to work on revenue management problems in the last five decades (Talluri and van Ryzin 2005).

The first mathematical approaches to airline revenue management were focused on single-leg problems. These models are generally grouped into two classes called static and dynamic. In short, the static models do not explicitly consider the dynamic nature of the arrivals of the requests over time, whereas the dynamic models are more responsive to the volatility in the demand as the end of the reservation period approaches and the seat capacity diminishes. Commonly used policies for capacity control center around booking limits or protection levels, nested allocations, and bid prices. In the case of a booking limit policy, the capacity of the aircraft is allocated to different fare classes. If a specific amount is reserved (protected) for different classes, then the control policy is based on protection levels. Nesting is a very intuitive notion for adjusting the booking limits or the protection levels. The main idea is to make the capacities for those lower-ranked (usually in terms of fares) classes available for higher-ranked (more expensive) classes. With a nested policy, a customer may book a seat as long as there is capacity available reserved for the requested-fare class or for the lower-fare classes. Therefore, a higher-fare class customer is never denied because of lack of capacity when there is available capacity for the lower-fare class customers. Consequently, the entire aircraft capacity is always available for the highest-fare class customers. Unlike setting the booking levels or the protection levels, a bid price is used to determine the threshold price (marginal revenue) of allocating a seat. This threshold price is then used for accepting or denying a request depending on whether the revenue from the request exceeds the threshold price. Because of their simplicity, the bid prices are used quite often in the industry for booking control. An effective implementation of this policy has to consider frequently updating the bid prices as time progresses and the available capacity changes. In this work, our focus shall be on determining the booking limits and the bid prices.

An airline typically controls the booking process over a large network of flights. As in the case of single-leg problems, it is relatively easy to write a stochastic dynamic programming model for the overall network problem. Unfortunately, this model suffers from the infamous curse of dimensionality because the state space becomes basically the Cartesian product of aircraft capacities in the flight network. Clearly, the resulting problem is intractable even for an airline running a relatively small-sized flight network. To overcome this difficulty, approximation methods based on decomposition and mathematical programming are proposed. The output of any one of these methods is then used to construct controls of various types, such as bid pricing and partitioned or (approximate) nested allocations.

Although leg-based decomposition approaches play an important role in network seat allocation, they focus on locally optimizing the booking, and hence, may undermine the network effects of shared aircraft capacities among multileg itineraries between origin-destination (OD) pairs. The main idea behind this type of decomposition is to represent the network problem as independent single-leg problems. To somehow consider the network effect, those single-leg problems are structured in such a way that they carry some information about the overall network. Among these leg-based decomposition methods, we can list dynamic programming decomposition (see §4 of this work for an implementation), prorated expected marginal seat revenue and OD factors method (Talluri and van Ryzin 2005).

We propose an approach that could be used to extend many single-leg models to the network setting by using an OD-based decomposition instead of a leg-based decomposition. A critical point in our proposal is based on the observation that whenever a capacity is allocated to an OD-pair (route), then the resulting problem over this particular route becomes a single-leg problem. Therefore, our approach consists of two stages: the optimal allocation of network capacities to OD-pairs and then, applying booking controls to different fare classes within each OD-pair (itineraries) after solving a single-leg problem. We not only concentrate on obtaining partitioned or nested booking limits, but also discuss that the proposed approach can be easily applied when bid-price control is exercised. When it comes to positioning our paper in the literature, the most noticeable work that is related to our approach is given by Curry (1990). In this seminal paper, Curry proposes an approach that is a combination of mathematical programming and nested allocation. Similar to our approach, his idea is also based on allocating capacities to the OD-pairs, but he then concentrates on the case where the fare classes within each OD-pair is nested. Curry did not report any computational results. Although his work is cited by many, only a few papers in the literature conducted a numerical study with this approach (Williamson 1992). We propose, on the other hand, a general framework where Curry's idea becomes a special case. We also complement our discussion with a numerical study on a network structure obtained from a major European airline through a consulting project[1] and benchmark the results against some well-known strategies used both in practice and in the literature. Our analysis along with the computational study suggest that Curry's idea can be extended to a broader setting when its generic nature is explicitly analyzed and supported by the numerical results as we do here. We also observe for moderate-size networks that the dynamic programming decomposition (DPD) is the most competitive method against the illustrative implementations of our framework and obtains higher average revenues than all considered strategies. However, DPD suffers from a high computational burden, particularly when the network size is large, and does not immediately yield to a general framework similar to the one proposed in this work.

1.1. Our Contributions

We propose and analyze a generic two-stage approach for solving large-scale airline network revenue management problems. We show that many single-leg solution methods can be used within the proposed framework. In fact, by using this framework, we also introduce a new model based on solving a dynamic single-leg problem at the first stage. The proposed framework also allows us to discuss immediate extensions of the proposed approach in other more recent network revenue management research areas, like robust optimization and customer choice models. We also establish that some special cases of the proposed approach are equivalent to several well-known network airline revenue management methods from

[1] Therefore, the authors are not permitted to disclose the name of the airline.

the literature. We conduct a thorough computational study on large-scale networks that are extracted from real-life data. To the best of our knowledge, we report results for networks that are much larger than those reported in the literature to this day. Our computational results indicate that two straightforward implementations of the proposed approach, coupled with a dynamic and static single-leg solution method, perform very well when compared against several standard benchmarking strategies in terms of collected revenue or computation time.

The rest of the paper is organized as follows: In §2 we give an overview of related literature. We introduce the proposed generic approach in §3, along with its immediate extensions to other airline revenue management models. In §4 we present our computational study. We conclude and discuss some future research directions in §5.

2. Review of Related Literature

The proposed approach in this paper relies on single-leg methods and extends those methods to a network setting. Therefore, we start with a brief account of the well-known methods in single-leg seat allocation literature. Then, we review the airline network revenue management literature. It is important to note that we do not explicitly consider overbooking, cancellations, customer choice behavior, or robustness. However, we do mention, particularly in §§3.3 and 5, some possible ways of extending the present work to those settings. We refer the interested reader to several recent single-leg and network papers and the references therein for a more complete overview (Aydın et al. 2013; Kunnumkal and Topaloglu 2011; Perakis and Roels 2010; van Ryzin and Vulcano 2008b).

The earliest work on single-leg problems is given in Littlewood (1972). He proposes a solution method for the case with only two fare classes. The idea behind his model is to equate the marginal revenues in each of the two fare classes. In his thesis, Belobaba (1987) extends this idea to a multiclass problem and introduces the expected marginal seat revenue heuristic for the general approach. Later, Brumelle and McGill (1993) work on this heuristic and obtain optimal policies for the multiclass problem. All of these solution methods are developed for the static problem, and they assume low-to-high fare class arrival pattern. Brumelle and McGill demonstrate that as long as the low-to-high fare class assumption holds, the static methods give the optimal solutions. Lee and Hersh (1993) address the dynamic problem by formulating it as a Markov decision process. In this model, the reservation period is divided into sufficiently small time intervals to allow only one arrival. In each period, a reservation request is accepted whenever its fare is higher than the expected marginal revenue of the seat. This work is refined by Liang (1999) and Lautenbacher and Stidham (1999). Whereas Liang reformulates the model in continuous time, Lautenbacher and Stidham combine the dynamic and static approaches under a common Markov decision process formulation. Recent studies in revenue management focus on the availability of information. Adaptive methods are used when there exists limited or no information about the demand. Most of these methods assume that there is access only to the samples from the demand distributions. They mainly compute the booking limits based on the past information, but also react to the possible inaccuracies related to the estimates of demand (Ball and Queyranne 2009; Birbil et al. 2009; Huh and Rusmevichientong 2006; Kunnumkal and Topaloglu 2009; Lan et al. 2008; van Ryzin and McGill 2000).

Glover et al. (1982) present the first mathematical programming approach to network airline revenue management. They formulate the problem as a minimum-cost network flow problem with side constraints. With this formulation, they were able to solve a large network problem by linear programming, but they did not consider the probabilistic passenger demand. In addition, their model is applicable only when the passengers are indifferent among the routes between OD-pairs. Wollmer (1986) proposes a binary programming formulation that incorporates demand stochasticity into the model by taking expected marginal seat revenues as coefficients in the objective function. Although he shows that this problem can be efficiently solved as a minimum-cost network flow problem, the model itself is impractical to use in large-scale networks. Curry (1990) develops a mathematical programming approach based on allocating capacities to the OD-pairs where the fare classes within each OD-pair is nested. The earliest work on bid-price controls is given by Simpson (1989), who proposes solving a linear programming model of the network problem and then using the optimal-dual variables to obtain the bid prices. Later in her thesis, Williamson (1992) gives a comprehensive study of similar mathematical programming formulations. Although the deterministic linear programming model of Simpson is now considered a classic in the field, this model is also criticized because it considers only the expected demands (which is slightly alleviated by frequently reevaluating the bid prices) and assumes an additive structure of the duals. Observing the lack of rigorous analysis in Simpson's and Williamson's works, Talluri and van Ryzin (1998) study in depth the bid-price controls and their asymptotic behavior. They show that the bid-price controls do not provide optimal solutions in general, but they perform well asymptotically. In a follow-up work, Talluri and van Ryzin (1999) propose a randomized linear programming model for computing the network bid prices.

Most of the standard bid prices are found by solving only the static network models. Bertsimas and Popescu (2003) introduce an algorithm based on approximate dynamic programming to capture the dynamics of the reservation process. Their approach is based on finding the total opportunity cost of the flight-leg capacities consumed by an itinerary request. For each itinerary, the marginal value of a flight-leg capacity is computed by taking the finite difference in the value functions. Although the numerical results demonstrate that their method outperforms the deterministic linear program, its computation time increases drastically as the network becomes larger. Adelman (2007) works with a linear programming representation of the dynamic programming formulation and aims at obtaining time-dependent (dynamic) bid prices. Recently, Tong and Topaloglu (2014) work on the model proposed by Adelman and improve its computational performance by an approximate linear programming approach. Topaloglu (2009), on the other hand, uses Lagrangean relaxation to compute bid prices that depend on the remaining capacity. In a short note, Talluri (2008) orders the bounds obtained by the deterministic linear programming model, the randomized linear programming model, and the models proposed by Adelman and Topaloglu. Using a stochastic approximation method, Topaloglu (2008) later computes the bid prices by visualizing the expected demand as a function of the bid prices and then works with sample path-based derivatives to search for a better set of bid prices. Surprisingly, Williamson (1992) observes that the elementary deterministic approximation methods based on average demands usually perform better than those advanced heuristics considering the probabilistic demand information. de Boer et al. (2002) investigate this phenomenon and conclude that this performance decrease is due to ignoring the nesting strategy. Although deterministic methods also do not consider nesting, the probabilistic methods suffer more from this negligence.

The virtual nesting method developed at American Airlines is one of the first decomposition approaches in airline revenue management (Smith, Leimkuhler, and Darrow 1992). This method distributes the total demand to a set of predefined virtual classes and then solves a multiclass single-leg problem. Therefore, the capacities of the flight legs can be administered independently. This decomposed structure allows nested control throughout the flight network. Two papers using simulation-based optimization methods that also investigate nesting over the network are written by Bertsimas and de Boer (2005) and van Ryzin and Vulcano (2008a). Bertsimas and de Boer propose a general framework to consider virtual classes and nesting as a general class of control strategies, which are parameterized by the set of protection levels on the network. Starting, then, with a set of nested booking limits, they use approximate dynamic programming to improve the booking limits. The model proposed by Bertsimas and de Boer (2005) considers discrete capacities and demand, and hence, their solution method does not guarantee convergence. van Ryzin and Vulcano (2008a), on the other hand, analyze the continuous version of the work of Bertsimas and de Boer and propose a faster and locally convergent algorithm. Another decomposition method combining mathematical programming methods and Markov decision processes is proposed by Cooper and Homem-de-Mello (2007). In a recent work, Zhang (2011) proposes an approximation approach to solve the network revenue management problem with customer choice. The approach is similar to dynamic programming decomposition and leads to new bounds on the optimal expected revenue.

3. OD-Based Decomposition Approach

In this section, we introduce our generic OD-based decomposition approach. The main idea behind the proposed approach is to separate the decision of allocating the flight-leg capacities to the OD-pairs from the decision of reserving seats to different fare classes within each OD-pair. In other words, the allocation of capacities to different OD-pair fare classes is carried out in two steps. In the first step, the capacity of a certain OD-pair is determined. Then, in the second step, this capacity is distributed to the fare classes of the same OD-pair. An important observation at this point is that the optimal distribution of the predetermined OD-pair capacity among its fare classes boils down to solving a single-leg problem.

To formalize our discussion, we start with a generic mathematical model. Suppose that there are S distinct OD-pairs on the network that belong to set $\mathcal{S} = \{1, \ldots, S\}$, and that there are J different flight legs coming from the set $\mathcal{J} = \{1, \ldots, J\}$. Moreover, we assume for each flight leg $j \in \mathcal{J}$ that the available seat capacity equals $C_j \in \mathbb{Z}_+$. Let

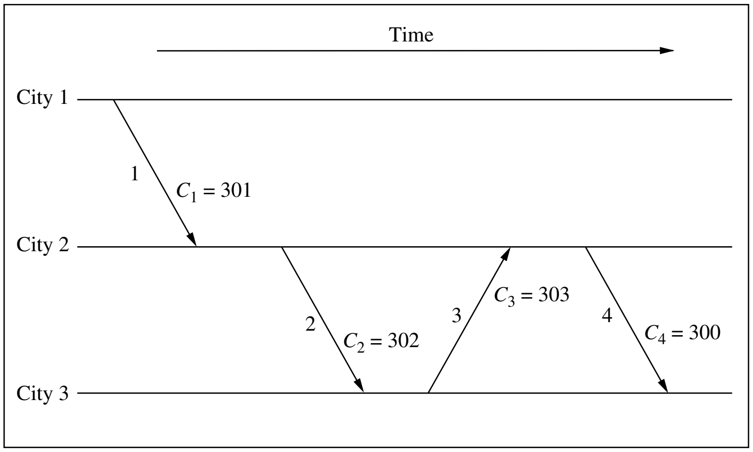
$$a_{js} = \begin{cases} 1, & \text{if flight leg } j \text{ is on OD-pair } s; \\ 0, & \text{otherwise.} \end{cases}$$

Then, the generic model is given by

$$\text{maximize} \quad \sum_{s=1}^{S} \phi_s(x_s), \tag{1}$$

$$\text{subject to} \quad \sum_{s=1}^{S} a_{js} x_s \leq C_j, \quad j \in \mathcal{J}, \tag{2}$$

$$x_s \in \mathbb{Z}_+, \quad s \in \mathcal{S}, \tag{3}$$



Possible OD-pairs

1
$1 \to 2$
$1 \to 2 \to 3$
$1 \to 2 \to 3 \to 4$
2
$2 \to 3$
$2 \to 3 \to 4$
3
$3 \to 4$
4

Figure 1 A Flight Network with Four Flight Legs and 10 Possible OD-Pairs
Note. For example, the bottleneck capacity on the sixth OD-pair ($2 \to 3$) is $B_6 = 302$.

where for a given x_s, the objective function $\phi_s(x_s)$ in (1) is itself an optimization problem that yields the optimal allocation of x_s capacities to different fare classes. Note that this is nothing but solving a single-leg problem with x_s capacities. We assume that $x_s \mapsto \phi_s(x_s)$ is a discrete concave function. Later, we will discuss that many well-known single-leg models that are proposed in the literature satisfy this assumption. The first set of constraints (2) ensures that the allocation to an OD-pair does not exceed the bottleneck flight-leg capacity on that particular OD-pair. This leads us to explicitly define the bottleneck capacity on each OD-pair $s \in \mathcal{S}$ by

$$B_s = \min_{j \in \mathcal{J}} \{C_j \mid a_{js} = 1\}. \qquad (4)$$

Notice that this definition of a bottleneck only applies to a selected OD-pair. That is, the bottleneck capacity is simply the minimum leg capacity on a multileg OD-pair. However, it should be noted that in a typical network problem, the capacity of a single leg is certainly shared by many OD-pairs, and this causes—as more commonly used—bottlenecks on the network. Figure 1 shows the flight legs and the OD-pairs on a small time-space network. It is also important to recall that a prominent advantage of using this kind of mathematical programming approach is to be able to use side constraints in the model like imposing an upper bound on the number of passengers for a particular OD-pair.

At first glance, the construction of the objective function (1) may require intensive computational effort because one needs to solve for each $s \in \mathcal{S}$ the single-leg problem $\phi_s(x_s)$ for all integer values of x_s from 1 to B_s. However, this is usually not required, because almost all existing solution methods for the single-leg models in the literature can produce an optimal solution for $\phi_s(B_s)$, which implicitly includes the optimal solutions for all the intermediate values $x_s \in \{1, \ldots, B_s - 1\}$. This is accomplished, for instance, through a dynamic programming algorithm, where the dimension of the state space is determined by the capacity. Hence, the optimal policy table already includes all optimal solutions from 1 to B_s. The complexity of constructing the objective function then depends mainly on the number of OD-pairs existing in the network rather than the summation of all bottleneck capacities over all OD-pairs. Naturally, this observation leads to significant savings in the computation time.

In the sequel, we shall use the solution of problem (1)–(3) to implement booking controls with partitioned or nested limits. Furthermore, we shall also consider the continuous relaxation of the same problem and obtain the dual-optimal variables corresponding to the flight capacities. This would then allow us to apply bid-price booking control strategies.

3.1. Two Illustrative Examples

We next consider one static and one dynamic single-leg model from the literature to exemplify the proposed OD-based decomposition approach. The static model assumes only probabilistic information about the total number of requests for a certain fare class corresponding to an itinerary. The dynamic model, on the other hand, assumes that the customers arrive over time and ask for different fare class tickets. The main purpose of our focus on these two particular models is twofold: First, these models belong to the broad classes known as static and dynamic models in the literature (Talluri and van Ryzin 2005). Moreover, we will later show that the first model is related to a well-known model from the literature. Second, these two models shall be used with different booking controls in our computational study, where we benchmark the proposed OD-based decomposition against several approaches that are frequently used for solving network problems in airline network revenue management.

Before we give the models, let us introduce the notation related to different fare classes within each OD-pair. There are I_s fare classes in each OD-pair $s \in \mathcal{S}$ that come from set $\mathcal{I}_s = \{1, \ldots, I_s\}$. The revenue of a fare class i seat in OD-pair s is given by r_{is}, and for each fixed OD-pair $s \in \mathcal{S}$, we assume without loss of generality that

$$0 < r_{1s} < r_{2s} < \cdots < r_{I_s s}, \tag{5}$$

where the no-sales class is simply represented by 0 with null revenue. If the integer decision variable x_{is} represents the number of reserved fare class i seats in OD-pair s, then for every $j \in \mathcal{J}$, the seat capacity restrictions (2) can also be written as

$$\sum_{s=1}^{S} a_{js} \sum_{i=1}^{I_s} x_{is} \leq C_j, \quad j \in \mathcal{J}.$$

Moreover, by relation (4) and the definition of x_{is}, we simply obtain

$$x_s = \sum_{i=1}^{I_s} x_{is} \leq B_s.$$

The static model assumes that we have probabilistic information about the aggregate demands for the OD-pair fare classes. Thus, the total demand $\mathbf{D}_{is}$ for fare class i seat in OD-pair s is a random variable. It is also assumed that each customer requesting a fare class i seat in OD-pair s is not buying a different fare class once fare class i in OD-pair s is sold out. In other words, the demands for different OD-pair fare classes are independent (Talluri and van Ryzin 2005). Then, it follows for each realization $\mathbf{D}_{is}(\omega)$ that $\min\{x_{is}, \mathbf{D}_{is}(\omega)\}$ will be the number of occupied fare class i seats in OD-pair s at departure. Thus, $\mathbb{E}[\min\{x_{is}, \mathbf{D}_{is}\}]$ is the expected number of occupied fare class i seats in OD-pair s, and we obtain the corresponding reformulation of problem (1)–(3) as

$$\text{maximize} \quad \sum_{s=1}^{S} f_s(x_s), \tag{6}$$

$$\text{subject to} \quad \sum_{s=1}^{S} a_{js} x_s \leq C_j, \quad j \in \mathcal{J}, \tag{7}$$

$$x_s \in \mathbb{Z}_+, \quad s \in \mathcal{S}, \tag{8}$$

where

$$f_s(x_s) = \max\left\{ \sum_{i=1}^{I_s} r_{is} \mathbb{E}[\min\{x_{is}, \mathbf{D}_{is}\}] \,\middle|\, \sum_{i=1}^{I_s} x_{is} \leq x_s,\, x_{is} \in \mathbb{Z}_+,\; i \in \mathcal{I}_s \right\}. \tag{9}$$

Note that (9) is just a single-leg problem with capacity x_s. Birbil et al. (2009) introduced a fast algorithm to solve this problem. When this algorithm is called to solve $f_s(B_s)$, it also produces, like dynamic programming, all optimal solutions for the intermediate values of x_s from 1 to B_s. Birbil et al. (2009) also showed for $s \in \mathcal{S}$ that $x_s \mapsto f_s(x_s)$ is discrete concave. Thus, the mathematical programming model (6)–(8) is in the structure required by the generic model (1)–(3). We will see in the next section that this is a well-known and frequently used model in the literature. Note that the optimal solution to this model gives partitioned booking limits. However, these limits can be used heuristically to obtain the booking limits that are nested by fares (cf. Williamson 1992). In fact, this will also be our approach in our computational study (§4).

Next we consider a dynamic model, where the demand for a fare class of an OD-pair is realized over time. In classical dynamic single-leg models of airline revenue management literature, it is quite common to assume that at each time period, at most one request arrives. Then, depending on the remaining capacity, the optimal strategy dictates whether to accept or reject the request (Lautenbacher and Stidham 1999). Using a similar approach, we also assume that there is at most one request for a fare class of an OD-pair at each period. Let T and $\hat{T}$ be the length of the reservation period and the number of discretization periods (epochs), respectively. If $g_s^t(x_s)$ is the expected optimal revenue for OD-pair s with available capacity x_s from period t to the departure in the last period, then the dynamic model within the proposed decomposition approach becomes

$$\text{maximize} \quad \sum_{s=1}^{S} g_s^1(x_s), \tag{10}$$

$$\text{subject to} \quad \sum_{s=1}^{S} a_{js} x_s \leq C_j, \quad j \in \mathcal{J}, \tag{11}$$

$$x_s \in \mathbb{Z}_+, \quad s \in \mathcal{S}. \tag{12}$$

To be precise, if we denote the random revenue generated at time t by ξ_t, then it is straightforward to obtain the dynamic programming recursion given by

$$g_s^t(x_s) = \mathbb{E}[\max\{\xi_t + g_s^{t+1}(x_s - 1), g_s^{t+1}(x_s)\}] \tag{13}$$

with the boundary condition

$$g_s^{\hat{T}}(x_s) = \begin{cases} \mathbb{E}[\xi_{\hat{T}}], & \text{if } x_s > 0; \\ 0, & \text{if } x_s = 0. \end{cases}$$

As x_s is fixed, the optimal value functions $g_s^1(x_s)$, corresponding to a single-leg dynamic programming problem, can be solved very efficiently. Furthermore, for each OD-pair $s \in \mathcal{S}$, the function $x_s \mapsto g_s^1(x_s)$ is discrete concave, and hence, problem (10)–(12) is again a special case of the generic model (1)–(3). It is important to note that the booking limits obtained by the optimal solution of $g_s^1(x_s)$ is already nested (Lautenbacher and Stidham 1999). The computational savings with this model are also worth mentioning. On one hand, the optimal policy table clearly includes the possible values of $g_s^1(x_s)$ for all $x_s \in \{1, \ldots, B_s\}$. On the other hand, note that the other dimension of the optimal policy table is time. Hence, this also leads to additional savings in computational effort, when one needs to reoptimize the problem at a time within the planning horizon. This is because the optimal policy table already includes for $t \in \{1, \ldots, \hat{T}\}$ the optimal values $g_s^t(x_s)$ for all $x_s \in \{1, \ldots, B_s\}$. Such reoptimizations are commonly carried out in practice, because it is advantageous to revise the booking control decisions as time elapses and the reservations materialize.

Although we call (10)–(12) a dynamic model, it should be emphasized that this model can only be considered as a partially dynamic model. That is, because the allocations to the OD-pairs are done at the beginning of the reservation period *before* the capacity of an OD-pair is depleted by the requests over time. A complete dynamic model, on the other hand, simultaneously considers the remaining capacities on each flight leg over time. Unfortunately, the complete dynamic model is intractable because keeping track of each remaining flight-leg capacity leads to an explosion of the state space, and hence, solving the complete dynamic model is not possible even for small-scale problems (Talluri and van Ryzin 2005).

The OD-based dynamic model given by (10)–(12) has another eminent role. As mentioned earlier, the complete dynamic model considers all possible combinations of capacity allocations. Therefore, allocating capacities to OD-pairs and then constructing optimal policies by solving the resulting model (10)–(12) is clearly a feasible solution to the complete dynamic model. Consequently, the optimal objective function value of the OD-based dynamic model gives a lower bound for the complete dynamic model. On the other hand, it is well known that the deterministic linear program gives an upper bound for the complete dynamic model; cf. Talluri (2008). Then, the difference between the objective function values of the deterministic linear programming model and the OD-based dynamic model gives the decision makers an estimate of the error by solving an approximation instead of solving the complete model. We shall revisit this remark in our computational study section and report some numerical results (see §4).

3.2. Properties of the Proposed Model

Note that a flight network in practice involves several hundreds of flight legs, and hence, the number of OD-pairs can become quite large. Therefore, it is customary to discuss the properties of the generic model given in (1)–(3). Furthermore, we observe that several well-known models from the literature are just special cases of our generic model. This section is devoted to discussing these properties and observations.

We start with a somewhat disconcerting result. Lemma 3.1 and its proof show that the problems resulting from our generic model are $\mathcal{NP}$-hard, even if the objective function is linear and the variables are binary.

Lemma 3.1. *Problem* (1)–(3) *is* $\mathcal{NP}$*-hard*.

Proof. Let $A \in \{0,1\}^{J \times S}$, $c \in \mathbb{R}^S$, and e be J dimensional vector of all ones. Then it is well known that the set-covering problem given by

$$\begin{aligned} \text{minimize} \quad & c^\intercal y, \\ \text{subject to} \quad & Ay \geq e, \\ & y \in \{0,1\}^S, \end{aligned}$$

is $\mathcal{NP}$-hard; see also Srinivasan (1999). By a simple change of variables $x = e - y$, we obtain

$$\text{maximize} \quad c^\intercal x \tag{14}$$

$$\text{subject to} \quad Ax \leq b, \tag{15}$$

$$x \in \{0,1\}^S, \tag{16}$$

where $b = Ae - e$. Clearly, problem (14)–(16) is also $\mathcal{NP}$-hard.

We will now show that problem (14)–(16) reduces polynomially to a special case of problem (1)–(3) with a linear objective function and binary variables. Take for each row in A, a hypothetical one-hour flight from city 1 to city 2. Assume that there is a one-hour interval between any two consecutive flights. We obtain the flight network illustrated in Figure 2. Note that any column in A consists of disconnected flight legs, all from city 1 to city 2. To make sure that each column in A can be represented as an OD-pair, we next duplicate flight network by adding one-hour auxiliary flights from city 2 to city 1, such that each new flight is a mirror flight of the flights associated with the original rows of A. Figure 3 shows the flight network with these auxiliary flights. By using these auxiliary flights in-between the original flights, we can always construct OD-pairs from the columns in A. Whenever an auxiliary flight is used for an OD-pair, the coefficient in the row that corresponds to the auxiliary flight becomes 1. If we denote the appended matrix by $\bar{A}$, then we obtain

$$\text{maximize} \quad c^\intercal x, \tag{17}$$

$$\text{subject to} \quad \begin{bmatrix} A \\ \bar{A} \end{bmatrix} x \leq \begin{bmatrix} b \\ b \end{bmatrix}, \tag{18}$$

$$x \in \{0,1\}^S. \tag{19}$$

We observe that problem (17)–(19) is a special case of a problem defined by (1)–(3). Therefore, we finish our polynomial reduction and conclude that the problem defined by (1)–(3) is $\mathcal{NP}$-hard. □

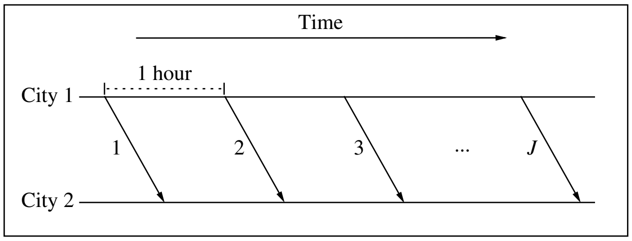


Figure 2 The Flight Network Used in the Proof of Lemma 3.1

Because problem (1)–(3) is difficult to solve, in the subsequent discussion we shall concentrate on the linear programming relaxation of this problem. Given the discrete concavity of its objective function, we can always replace the objective function by a piecewise-linear concave function and relax the integrality constraints. Then, the resulting model simply becomes a linear programming problem. However, the optimal solution of this linear programming model may be noninteger, as formally implied by Lemma 3.1. To complement this discussion, we show in Example 3.1 that even for a very simple flight network with linear objective function, the optimal solution of the approximate problem can be fractional; see also Glover et al. (1982) and Bertsimas and de Boer (2005) for a discussion on network flow formulations that yield integer optimal solutions.

Example 3.1. Consider the flight network shown in Figure 1 with only three OD-pairs, $1 \to 4$, $1 \to 2$, and $2 \to 3 \to 4$, respectively. If we assume for simplicity that the objective function of problem (1)–(3) is given by a simple linear function, then the linear programming relaxation of this problem becomes:

$$\begin{array}{lllll} \text{maximize} & x_1 & +x_2 & +x_3, & \\ \text{subject to} & x_1 & +x_2 & & \le 301, \\ & & x_2 & +x_3 & \le 302, \\ & & & x_3 & \le 303, \\ & x_1 & & +x_3 & \le 300, \\ & \multicolumn{4}{l}{x_1 \ge 0, \quad x_2 \ge 0, \quad x_3 \ge 0.} \end{array}$$

The *unique* optimal solution of this problem is given by

$$(x_1^*, x_2^*, x_3^*) = (149.5, 151.5, 150.5).$$

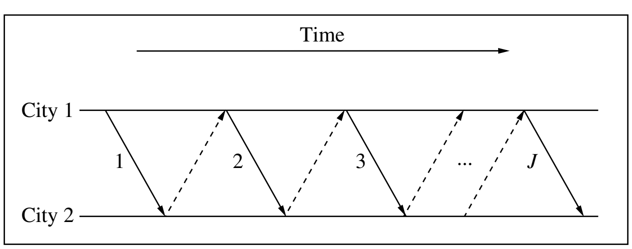


Figure 3 The Flight Network After Introducing the Auxiliary Flights

Because we propose to solve the linear programming relaxation of problem (1)–(3), we can then simply round down the noninteger allocation x_s to obtain a feasible solution. Let x_s^* be the optimal solution of problem (1)–(3) and $\hat{x}_s$ be the optimal solution of its LP relaxation. Clearly, the rounded-down solution $\bar{x}_s\colon = \lfloor \hat{x}_s \rfloor$ is feasible for problem (1)–(3). If we denote the objective function values by $v^* = \sum_{s=1}^{S} \phi_s(x_s^*)$, $\hat{v} = \sum_{s=1}^{S} \phi_s(\hat{x}_s)$, and $\bar{v} = \sum_{s=1}^{S} \phi_s(\bar{x}_s)$, then we have $\bar{v} \le v^* \le \hat{v}$. As we will see in our computational study, the gap $\hat{v} - \bar{v}$ turns out to be quite tight for the two illustrative examples.

A very important advantage of using the linear programming relaxation of problem (1)–(3) is the information gained by its dual. In other words, when this relaxation is solved to optimality, there will be an optimal-dual variable associated with each flight leg. Using this optimal-dual solution, we can also discuss the shadow prices associated with the changes in the capacities of the flight legs. This is nothing but using the bid-price control as a booking strategy. A straightforward implementation is then to grant a seat to an OD-pair request when the revenue gained from the request exceeds the sum of the optimal-dual variables corresponding to those flights traversed by the requested OD-pair. Consequently, the difficulty with obtaining an integer solution to (1)–(3) also disappears. As with most static models, the bid-price approach and its performance have been studied before. However, for other models embedded in our framework, like our dynamic model (10)–(12), this type of bid-price control leads to interesting results and observations, as we elaborate in §4.

The proposed generic model also has some close ties with several previous works in the literature. We start with the famous deterministic linear programming model considered initially by Glover et al. (1982), Wollmer (1986), Simpson (1989), and Williamson (1992), and show that it is a special case of our proposed model (1)–(3). We shall then revisit our static model (6)–(8) and present that this model is also equivalent to another model discussed in Wollmer (1986), Williamson (1992), and Talluri and van Ryzin (2005).

Suppose the expected demand $d_{is} \in \mathbb{Z}_+$ for fare class i seats in OD-pair s is given. Then $\min\{x_{is}, d_{is}\}$ is the number of occupied fare class i seats in OD-pair s at the departure time. Then, we obtain

$$\text{maximize} \quad \sum_{s=1}^{S} h_s(x_s), \tag{20}$$

$$\text{subject to} \quad \sum_{s=1}^{S} a_{js} x_s \le C_j, \quad j \in \mathcal{J}, \tag{21}$$

$$x_s \in \mathbb{Z}_+, \quad s \in \mathcal{S}, \tag{22}$$

where

$$h_s(x_s) = \max\left\{\sum_{i=1}^{I_s} r_{is}\min\{x_{is}, d_{is}\} \,\middle|\, \sum_{i=1}^{I_s} x_{is} \le x_s, x_{is} \in \mathbb{Z}_+, \right.$$
$$\left. i \in \mathcal{I}_s \right\}. \quad (23)$$

As shown by Birbil et al. (2009), the single-leg problem (23) enjoys an analytical solution for fixed x_s and it is trivial to see that the optimal objective function value is concave in terms of x_s. It is clear then that the objective function (20) also becomes discrete concave, and hence, this formulation fits nicely into the generic framework given by (1)–(3). As a side note, we also remark that the optimal objective function value of problem (20)–(22) can serve as an upper bound for the static problem (6)–(8) because of the well-known inequality

$$\mathbb{E}[\min\{x_{is}, \mathbf{D}_{is}\}] \le \min\{x_{is}, d_{is}\},$$

where $\mathbf{D}_{is}$ denotes the random demand for fare class i in OD-pair s with $\mathbb{E}[\mathbf{D}_{is}] = d_{is}$.

As discussed in Wollmer (1986), de Boer et al. (2002), Talluri and van Ryzin (2005), one can give the following integer programming formulation to the same problem:

$$\text{maximize} \quad \sum_{s=1}^{S}\sum_{i=1}^{I_s} r_{is}\min\{x_{is}, d_{is}\}, \quad (24)$$

$$\text{subject to} \quad \sum_{s=1}^{S} a_{js}\sum_{i=1}^{I_s} x_{is} \le C_j, \quad j \in \mathcal{J}, \quad (25)$$

$$x_{is} \in \mathbb{Z}_+, \quad s \in \mathcal{S}, i \in \mathcal{I}_s. \quad (26)$$

An equivalent integer programming formulation is also given by

$$\text{maximize} \quad \sum_{s=1}^{S}\sum_{i=1}^{I_s} r_{is}x_{is}, \quad (27)$$

$$\text{subject to} \quad \sum_{s=1}^{S} a_{js}\sum_{i=1}^{I_s} x_{is} \le C_j, \quad j \in \mathcal{J}, \quad (28)$$

$$x_{is} \le d_{is}, \quad s \in \mathcal{S}, i \in \mathcal{I}_s, \quad (29)$$

$$x_{is} \in \mathbb{Z}_+, \quad s \in \mathcal{S}, i \in \mathcal{I}_s. \quad (30)$$

The following lemma shows that all three preceding mathematical programming models are equivalent. This may also be a noteworthy result from a computational point of view, because after constructing the objective function, which can be done by solving (23) analytically, our model (20)–(22) has a fewer number of variables and the same number of constraints as in (27)–(30). This gain in computation time may even increase when frequent reoptimizations are carried out because the flight capacities, and consequently the bottleneck capacities, decrease as time elapses. Moreover, in real-life instances the problems may involve enormous numbers of itineraries, and hence, it may be viable to select our formulation, because its memory requirements are better than those of the deterministic linear program.

Lemma 3.2. *Let $(x_s^*)_{s\in S}$ be an optimal solution of problem* (20)–(22). *If $(x_{is}^*)_{i\in\mathcal{I}_s}$ is an optimal solution of* (23) *for $x_s = x_s^*$, then it is also an optimal solution of* (24)–(26). *Moreover, the optimal objective values of the problems* (24)–(26) *and* (27)–(30) *are the same, and (x_{is}^*) is also an optimal solution of* (27)–(30).

Proof. We start with the first part. Since for each s the vector $(x_{is}^*)_{i\in\mathcal{I}_s}$ is integer and an optimal solution of problem (23) for integer x_s^* is again integer, it must hold by definition that the inequality in (23) is binding. This means that $x_s^* = \sum_{i=1}^{I_s} x_{is}^*$. Thus, (x_{is}^*) is feasible for problem (24)–(26). This shows that $v \le v_0$, where v and v_0 denote the optimal objective values of (20)–(22) and (24)–(26), respectively. Also, by the definition of $(x_{is}^*)_{1\le i\le I_s 1\le s\le S}$, we obtain that

$$\sum_{s=1}^{S}\sum_{i=1}^{I_s} r_{is}\min\{x_{is}^*, d_{is}\} = \sum_{s=1}^{S} f_s(x_s^*).$$

Moreover, for any feasible solution (x_{is}) of problem (24)–(26), we obtain that $x_s := \sum_{i=1}^{I_s} x_{is}$ is feasible for (20)–(22) and by the definition of $f_s(\cdot)$, that

$$\sum_{s=1}^{S}\sum_{i=1}^{I_s} r_{is}\min\{x_{is}, d_{is}\} \le \sum_{s=1}^{S} f_s(x_s).$$

This shows the first part.

In the second part, notice that the feasible region $\mathcal{F}_1$ of optimization problem (27)–(30) is a subset of the feasible region $\mathcal{F}_0$ of optimization problem (24)–(26), and for every $(x_{is}) \in \mathcal{F}_1$ it holds that $x_{is} = \min\{x_{is}, d_{is}\}$. Thus, we obtain $v_1 \le v_0$, where v_1 is the optimal objective function value of problem (27)–(30). Moreover, if (x_{is}^*) is an optimal solution of optimization problem (24)–(26), then clearly the vector (x_{is}^*) with $x_{is}^* = \min\{x_{is}^*, d_{is}\}$ is feasible for problem (27)–(30) and both objective values coincide. This shows that $v_0 \le v_1$, and so, $v_0 = v_1$. By this and the relation $\mathcal{F}_1 \subseteq \mathcal{F}_0$, the last part becomes obvious. □

As we mentioned previously, the continuous relaxation of (27)–(30) is known as the deterministic linear programming problem. In particular, the majority of the models proposing different bid-price control strategies stems from this relaxation. Therefore, this model is frequently used for benchmarking in the literature. Using a similar argument as in Lemma 3.2, one can easily show that the optimal solution of the deterministic linear programming problem can also be obtained by solving the continuous relaxation of problem (20)–(22). As before, the continuous relaxation of subproblem (23) has an analytical solution. Thus, the possible computational savings that we discussed previously are still applicable. To be precise, the number of variables in the standard deterministic linear program is equal to the number of itineraries (the total number of classes in all OD-pairs), and the number of constraints is equal to the number of flight legs in (28) plus the number of itineraries in (29). The number of variables in the continuous relaxation of our model (20)–(22), on the other hand, is equal to the number of OD-pairs. After applying a straightforward reformulation to the piecewise-linear objective function, the number of variables in the resulting linear program is doubled and the number of constraints is increased by the number of itineraries. However, we should note that the constraints designated by (29) are just upper-bound constraints, whereas the same number of constraints coming from the reformulation of the objective function of our model corresponds to the linear pieces of each $x_s \mapsto h_s(x_s)$, $s \in \mathcal{S}$. The upper-bound constraints can be effectively dealt with using a bounded simplex method implementation. The same advantage does not immediately apply to the constraints added by our method. Even so, there could be a remedy for this drawback if one implements a specialized simplex method such as the one proposed by Fourer (1992), who observed that the additional constraints are not needed and the solution time can be decreased significantly when compared against a standard simplex method.

It is well known that the deterministic linear programming problem does not necessarily yield optimal integer solutions (also demonstrated by Example 3.1). Nonetheless, the following lemma shows that if the capacity and demand values are integer, then at the optimal solution of the LP relaxation of problem (27)–(30), there will be at most one fare class on each OD-pair that has noninteger allocation. This result illustrates why the LP relaxation optimal objective function value is frequently quite close, or even equal, to the integer optimal objective function value.

Lemma 3.3. *Suppose that d_{is} for all $i \in \mathcal{I}_s$, $s \in \mathcal{S}$ and C_j for all $j \in \mathcal{J}$ are integer. Let (x_{is}^*) be the optimal solution of the LP relaxation of problem* (27)–(30). *Then, for each OD-pair $s \in \mathcal{S}$, there will be at most one fare class $i_s \in \mathcal{I}_s$ such that $x_{i_s s}^*$ is noninteger.*

Proof. Lemma 3.2 establishes the equivalence between problems (20)–(22) and (27)–(30). Therefore, if we obtain a noninteger solution for an OD-pair in (20)–(22), then the noninteger part will be allocated to only one fare class as a direct consequence of relation (5). □

In his seminal work, Wollmer (1986) discusses a model that boils down to the following integer programming formulation:

$$\text{maximize} \quad \sum_{s=1}^{S}\sum_{i=1}^{I_s} r_{is}\mathbb{E}[\min\{x_{is}, \mathbf{D}_{is}\}], \tag{31}$$

$$\text{subject to} \quad \sum_{s=1}^{S} a_{js}\sum_{i=1}^{I_s} x_{is} \le C_j, \quad j \in \mathcal{J}, \tag{32}$$

$$z_{is} \in \mathbb{Z}_+, \quad i \in \mathcal{I}_s, s \in \mathcal{S}. \tag{33}$$

The continuous relaxation of this model is also considered in Talluri and van Ryzin (1998) and de Boer et al. (2002). As shown in Lemma 3.4, problems (6)–(8) and (31)–(33) are equivalent.

Lemma 3.4. *If $(x_s^*)_{s\in S}$ is an optimal solution of problem* (6)–(8), *then it follows that $(x_{is}^*)_{i\in\mathcal{I}_S, s\in\mathcal{S}}$ with for each $s \in \mathcal{S}$ the vector $(x_{is}^*)_{i\in\mathcal{I}_s}$ is an optimal solution of the optimization problem associated with $g_s(x_s^*)$ is an optimal solution of* (31)–(33).

Proof. See the first part of Lemma 3.2. □

Wollmer (1986) in fact considered a binary programming formulation of problem (31)–(33), where the decision variable x_{is} is replaced by the summation of binary variables x_{isl}. Observe that the function $F_{is}\colon \mathbb{Z}_+ \to R$ given by

$$F_{is}(n) := \mathbb{E}[\min\{n, \mathbf{D}_{is}\}], \tag{34}$$

satisfies

$$F_{is}(n) - F_{is}(n-1) = \mathbb{P}\{D_{ik} \ge n\}. \tag{35}$$

Hence, it is decreasing in n and so the function F_{is} is a discrete concave function on $\mathbb{Z}_+$. Using relation (35) and the fact that $F_{is}(\cdot)$ is a discrete concave function, we obtain the binary programming model equivalent to problem (31)–(33):

$$\text{maximize} \quad \sum_{s=1}^{S}\sum_{i=1}^{I_s} r_{is}\sum_{l=1}^{B_s} x_{isl}\mathbb{P}\{\mathbf{D}_{is} \ge l\}, \tag{36}$$

$$\text{subject to} \quad \sum_{s=1}^{S} a_{js}\sum_{i=1}^{I_s}\sum_{l=1}^{B_s} x_{isl} \le C_j, \quad j \in \mathcal{J}, \tag{37}$$

$$x_{isl} \in \{0,1\}, \quad s \in \mathcal{S}, i \in \mathcal{I}_s, l = 1, \ldots, B_s. \tag{38}$$

The same optimization problem is also discussed in de Boer et al. (2002) and Talluri and van Ryzin (2005). This binary problem has the same optimal objective value as (31)–(33), and for (x_{isl}^*) being an optimal solution of (36)–(38), the corresponding optimal solution of problem (31)–(33) is given by

$$x_{is}^* = \sum_{l=1}^{B_s} x_{isl}^*.$$

Clearly, problem (36)–(38) has a huge number of variables, and in general it is quite difficult to transform the feasible solution of the original problem. Therefore, the binary programming formulation is not very suitable from a computational point of view. With our formulation (6)–(8), however, subproblem (9) can be solved very efficiently. Consequently, the computation times are reduced significantly like the OD-based formulation of the deterministic linear programming problem.

3.3. Other Applications of the Proposed Model

It is important to underline that the structure of the proposed model (1)–(3) allows us to carry many different single-leg models to a network setting. Recall that the main requirement in the proposed model is to have a discrete concave objective function. This requirement is met whenever the optimal objective function value of the underlying single-leg problem is concave with respect to the allocated capacity. Fortunately, an ample number of well-known models from the literature satisfy this condition.

One of the definitive references on revenue management is written by Talluri and van Ryzin (2005). In Chapter 2 of their work, Talluri and van Ryzin summarize single-resource optimal capacity control for static, dynamic, and customer choice models. The majority of these models are analyzed by considering the opportunity cost through the structure of a value function that measures the optimal expected revenue as a function of the remaining capacity. A key result in such an analysis is showing that the opportunity costs are nonincreasing. This result is quite intuitive, because one does not expect that the marginal revenues obtained by allocating more resources shall increase unceasingly. As a direct consequence, these diminishing marginal values imply that the optimal objective function values of these single-resource models are discrete concave with respect to the capacity. Therefore, not only the static and dynamic models, but even some of the customer choice models, used in single resource control can be used for solving multiple-resource (network) models in practice by incorporating them into the objective function of the proposed model (1)–(3).

In their recent work, Birbil et al. (2009) study the robust versions of the standard static and dynamic single-leg problems, where they consider the inaccuracies associated with the probability distributions of the customer demands. Their robust optimization models basically couple the standard (nonrobust) models with analytically solvable problems, and hence, the overall problem structure does not change. Therefore, the resulting robust static and dynamic models remain to have discrete concave optimal objective function values with respect to the flight-leg capacities. This immediately implies that the robust single-leg models can be used within our generic model (1)–(3). Thus, a robust optimization approach, based on the work of Birbil et al. (2009), can also be proposed for the network airline revenue management problem.

4. Computational Study

Although the main motivation of this work is to introduce a general framework, we also want to show in this section that even straightforward implementations within the proposed framework may yield competitive solution methods. Next, we present a set of numerical experiments and compare our two illustrative models against several well-known network revenue management models from the literature.

We use a real network structure obtained from a major European airline. These data include the actual flight legs, their capacities, and the OD-pairs. The airline also provided the ticket price for each fare class. However, these data were stored per flight leg. Therefore, we had to generate the fares for OD-pair classes by taking into account the given ticket prices on the flight legs within each OD-pair. As shown in Table 1, we worked on subnetworks with varying sizes extracted from the overall network. The largest subnetwork that we use includes 5,000 OD-pairs. To the best of our knowledge, this is by far the largest number of OD-pairs among those reported in the airline revenue management literature. The largest network that we are aware of is studied by van Ryzin and Vulcano (2008a), and this network includes only 62 flight legs and contains in total 1,844 itineraries. Note that this figure corresponds roughly to the number of OD-pairs multiplied by the average number of fare classes within OD-pairs. For our largest subnetwork, the same number turns out to be 45,182.

To give a further idea of our network structure, Tables 2 and 3 show for each subnetwork the breakdown of the total number of OD-pairs in terms of the number of flight legs and the number of fare classes, respectively. Table 2 shows that the longest OD-pair consists of five flight legs, and the majority of the OD-pairs are constructed with one to three flight legs. As Table 3 illustrates, the total number of OD-pairs is distributed almost uniformly among different numbers of fare classes. The exceptions to this pattern may be the extreme cases of three and 15 fare classes.

Table 1 **The Numbers of Itineraries and Flight Legs in the Subnetworks**

	Network size*			
	100	500	1,000	5,000
Number of flight legs	119	428	583	678
Number of itineraries	869	4,470	9,170	45,182

*Total number of OD-pairs in the subnetwork.

Table 2 Breakdown of the Total Number of OD-Pairs in Terms of Number of Flight Legs

	Number of flight legs				
Network size	1	2	3	4	5
100	65	30	5	—	—
500	280	178	40	2	—
1,000	428	455	105	12	—
5,000	631	3,353	934	80	2

4.1. Simulation Setup

Lacking the actual demand distributions, we simulate the arrival of reservation requests in a planning horizon of length T. We assume that the requests for OD-pair $s \in \mathcal{S}$ arrive according to a homogeneous Poisson process with rate λ_s. Given that a request for OD-pair s arrives at time t, this request is for fare class i with probability $p_{is}(t)$. Clearly, $p_{is}(t) \geq 0$ and $\sum_{i=1}^{I_s} p_{is}(t) = 1$ for all $s \in \mathcal{S}$. Thus, each arriving request is labeled as a certain fare-class request by using the multinomial probabilities that vary over time, $p_{is}(t)$, $i = 1, \ldots, I_s$, $0 \leq t \leq T$. Assuming that in reality the lower-fare class requests arrive more frequently in the early periods than the higher-fare classes, we set the multinomial probabilities as

$$p_{is}(t) = \frac{(v_{is}^T - v_{is}^0)t + Tv_{is}^0}{\sum_{j=1}^{I_s}(v_{js}^T - v_{is}^0)t + Tv_{js}^0}, \quad i = 1, \ldots, I_s,$$

where $0 \leq v_{I_s s}^0 < v_{I_s-1s}^0 < \cdots < v_{1s}^0$, and $0 \leq v_{1s}^T < v_{2s}^T < \cdots < v_{I_s s}^T$ are predefined parameters. This way of setting the multinomial probabilities complies with the desired demand pattern. An illustration is given in Figure 4.

The booking requests arriving over time are simulated as follows: We first generate the arrival time of a booking request for each OD-pair during planning horizon T. The minimum time among the booking request times determines the arriving OD-pair and the next event time. Then, using the multinomial probabilities, we find the fare class of the request and apply the booking policy of interest. After the number of reservations for all fare classes in that OD-pair are updated, the simulation continues with determining the next event time. To test the performances of the booking policies against varying arrival intensities, we also use the load factor parameter (ρ) and tie the arrival rate of an OD-pair s to the given load factor. First, we come up with an estimate of arrival rate μ_j for each flight leg $j \in \mathcal{J}$ by

$$\mu_j = \rho \frac{C_j}{TS_j},$$

where S_j denotes the number of OD-pairs using flight leg j. Then, the arrival rate for an OD-pair $s \in \mathcal{S}$ simply becomes

$$\lambda_s = \frac{\sum_{j=1}^{J_s} \mu_j}{J_s},$$

where J_s is the number of flight legs that are used by OD-pair s.

4.2. Benchmarking Strategies

In the network revenue management literature and also in practice, the deterministic linear program (27)–(30) and its variations are favored as viable solution methods for network problems because of their simplicity and speed. In our benchmarking study, we also use two of those methods. Unless resolved frequently, the bid prices obtained by the mathematical programming based methods do not depend on time and inventory. To compare against bid-pricing approaches, we also use a dynamic programming leg-based decomposition heuristic. All these benchmarking strategies, as well as our illustrative methods, are detailed as follows:

(i) Deterministic Linear Program (DLP): This strategy solves the model (27)–(30) for approximating the optimal total expected revenue as well as computing the bid prices. When a request for an itinerary arrives, the summation of the optimal-dual variables corresponding to those flight legs used by the itinerary is used as the bid price for accepting or rejecting the request; cf. Talluri and van Ryzin (1998).

(ii) Finite Differences on Deterministic Linear Program (FDD): This strategy focuses on the total opportunity cost of the flight-leg capacities consumed by an OD-pair request (itinerary). We first compute the optimal expected revenue with the current levels of the capacities by solving (27)–(30). Then for each OD-pair a new linear program is set up, where the capacities of those flight legs used in the OD-pair is decreased by one. This reflects the situation of allocating one seat

Table 3 Breakdown of the Total Number of OD-Pairs in Terms of Number of Fare Classes

	Number of fare classes												
Network size	3	4	5	6	7	8	9	10	11	12	13	14	15
100	6	11	7	3	14	9	9	8	9	5	5	11	3
500	27	34	47	38	44	39	42	41	41	55	36	37	19
1,000	45	66	81	77	83	76	86	92	84	101	89	72	48
5,000	182	416	426	405	435	393	441	409	434	401	446	400	212

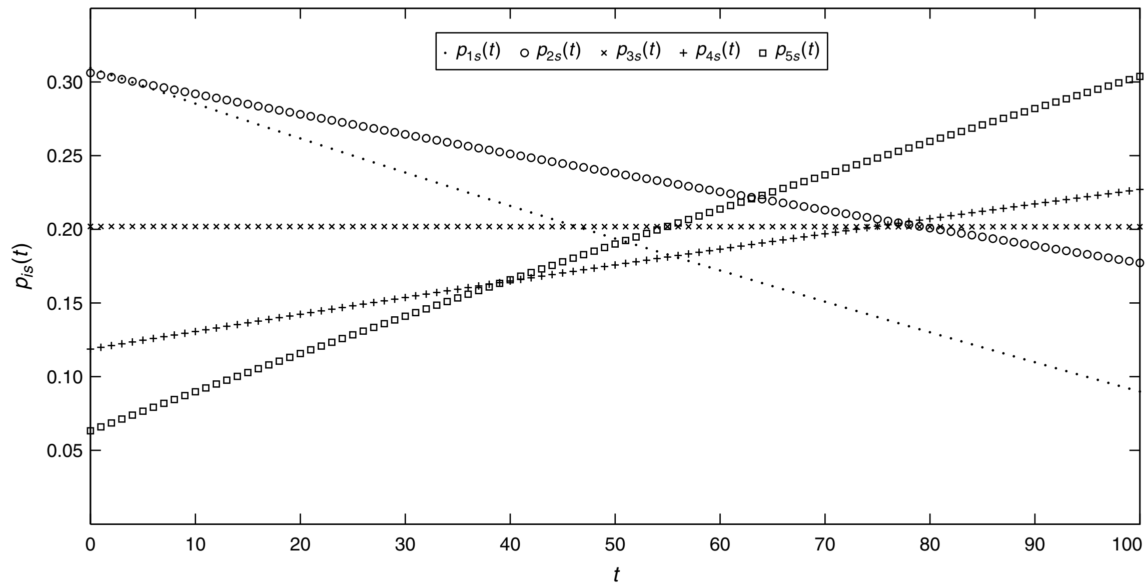


Figure 4 **An Example of the Changes in Multinomial Probabilities Over Time** ($I_s = 5$)

to a request from that OD-pair. After solving this new linear program, the difference between the optimal expected revenues of the problem with the original capacities and the new problem becomes the threshold value. We accept an arriving OD-pair request if the fare price of this itinerary is greater than the calculated threshold value. We repeat these steps for all OD-pairs; cf. Bertsimas and Popescu (2003). In this strategy, a separate linear program is solved for each OD-pair.

(iii) OD-based Decomposition—The Static Models (ODB-STA, ODB-STB, and ODB-STD): These three strategies are related to our first illustrative model (6)–(8) using only the static demand information. The subproblem (9) is solved by the algorithm proposed by Birbil et al. (2009). In ODB-STA we obtain the partitioned booking limits and use them in the simulation after nesting them. ODB-STB uses the optimal-dual variables to compute the bid prices. The third strategy, ODB-STD, is similar to FDD, where we resolve the problem for each OD-pair after reducing the associated flight-leg capacities by one. Then, the difference in the objective function values is used as the threshold value for accepting or rejecting an itinerary request.

(iv) OD-based Decomposition—The Dynamic Models (ODB-DPA, ODB-DPB, and ODB-DPD): The remaining three strategies are variations of our second model (10)–(12), where the objective function is constructed by solving the dynamic programming recursion given by (13). After solving the model, ODB-DPA uses the optimal nested booking limits as a control policy, whereas ODB-DPB stores the optimal-dual variables to obtain the bid prices. Like ODB-STD, the final strategy ODB-DPD works with the total opportunity cost of allocating a seat to an itinerary request.

(v) Dynamic Programming Decomposition (DPD): This is a decomposition heuristic based on using displacement-adjusted revenues to decompose the network into a series of single-leg problems (Talluri and van Ryzin 2005, §3.4.4). The decomposition starts with a set of fixed bid prices for each flight leg. As also suggested by Talluri and van Ryzin (2005), we solve DLP to obtain these fixed bid prices. The displacement-adjusted revenue for an itinerary using a particular flight leg is evaluated by subtracting the fixed bid prices of the other flight legs used by the same itinerary from the itinerary's actual revenue. Then, these displacement-adjusted revenues for a particular flight leg are used to solve a single-resource dynamic programming model given by the recursion (13). Note that in this method, a single-leg dynamic model is solved for a much larger number of classes than solving it for an OD-pair. This is because a vast number of OD-pairs may share the same flight leg, and consequently, all of their classes need to be considered in the dynamic programming recursion. This heuristic requires a very fine discretization of the reservation period, since the dynamic model (13) assumes that there is at most one arrival (itinerary request) in one period; see also Talluri and van Ryzin (2005, §2.5).

To consider the effects of reoptimization, we also divide the planning horizon into K equal segments and revise the strategies at the beginning of each segment with the updated capacities. In the sequel, we refer to ODB-STA and ODB-DPA as the allocation-based strategies, whereas all the remaining ones are called the bid-price strategies.

4.3. Numerical Results

In our simulation experiments, the parameters for the multinomial probabilities, v_{is}^T and v_{is}^0, are uniformly distributed from the interval $(0, 3I_s)$ and then sorted in the required order for each $s \in \mathcal{S}$ and $i \in \mathcal{I}_s$. Our experimental design is based on various factors such as the network size, the load factor (ρ), and the number of reoptimization segments in a solution period (K). We test the benchmark strategies with respect to load factor values $\rho \in \{1.0, 1.2\}$ to represent regular and high loads. The last parameter set comes from the number of segments in the planning horizon $K \in \{1, 5, 10\}$. The average revenues and computation times obtained by different benchmark strategies are reported over 25 simulation runs. We take the reservation period length as $T = 100$. For OD-based dynamic strategies (ODB-A, ODB-B, and ODB-D), the discretization mesh size is set to $5.0e-2$ for moderate-size networks and $1.0e-1$ for large-size networks. For DPD, the mesh size is further divided into smaller periods to ensure that there is at most one arrival in each epoch.

Figures 5 and 6 show our results for moderate-size networks with 100 and 500 OD-pairs, respectively. Both figures demonstrate that when reoptimization is not applied ($K = 1$), the allocation-based strategy, ODB-DPA, and the decomposition heuristic, DPD, outperform all other strategies. The average revenue obtained by DPD is marginally better than that of ODB-DPA. As expected, applying reoptimization has a significant effect on those remaining methods that rely on bid-prices. However, the same effect on allocation-based strategies is not that substantial. When reoptimizations are carried out ($K = 5$ and $K = 10$) and the network becomes larger, OD-based decomposition methods using bid prices, ODB-STB, ODB-STD, ODB-DPB, and ODB-DPD, along with DLP and FDD, obtain average revenues that are close to those obtained by ODB-DPA and DPD. In fact, as the bar plots for $K = 5$ and $K = 10$ in Figure 6 show, ODB-DPD and ODB-DPB obtain even slightly higher revenues than that of ODB-DPA, but still less than DPD. The load factor, in general, does not seem to have a large impact on the lineup of the strategies but it does slightly boost the performance of ODB-DPA as this dynamic model responds to the increase in the number of customer requests better than the other OD-based decomposition strategies. As Figure 5 illustrates, for the network with 100 OD-pairs, the other allocation-based strategy, ODB-STA, outperforms only two strategies, DLP and FDD, when reoptimization is abandoned ($K = 1$). Unfortunately, when the network size increases, its performance deteriorates even when $K = 1$, and it becomes the least effective strategy. When we look at the results for the moderate-size networks, we observe that there is almost no difference between using bid prices or finite differences for OD-based static (ODB-STB versus ODB-STD) or dynamic (ODB-DPB versus ODB-DPD) strategies. Surprisingly, the difference between DLP and FDD, on the other hand, is more striking.

We report in Figure 7 the average computation times obtained with different strategies on a semilogarithmic plot (the average computation times in the vertical axis are given in logarithmic scale). Note that ODB-STA and ODB-STB, and similarly, ODB-DPA and ODB-DPB, take the same computation time. Thus, we give only one figure for each pair. The first thing one notices in Figure 7 is that DPD takes, by far, the largest computation time among all strategies. This is because the single-leg dynamic programming models, which have to be solved for each flight leg in the DPD method, require much finer discretization because the number of fare classes increases tremendously, especially when the network size increases. In ODB-based approaches, the number of fare classes is at most 15 for an OD-pair (see Table 3). However, the total number of fare classes sharing a flight leg goes up to several hundreds as the network size increases simply because a flight leg is shared by too many OD-pairs. Therefore, each dynamic program that needs to be solved for the DPD method takes a very long time, and the memory requirement is far more than that of a dynamic model solved for the OD-based methods. Furthermore, the DPD method starts with a guess of initial bid prices to evaluate the displacement-adjusted revenues (Talluri and van Ryzin 2005, §3.4.4). To serve this purpose, we have solved DLP and used its dual-optimal solution. Thus, when we apply reoptimization, all those single-leg dynamic programs used in the DPD method have to be solved once again. However, when it comes to the ODB-DPA(B) method, the single-leg dynamic programs are solved once at the very beginning, and never again. This is also clear from Figure 7; the increase in the computation time for ODB-DPA(B) is insignificant when reoptimizations are carried out, but applying reoptimization increases the computation time considerably for the DPD method. Among the OD-based strategies, the dynamic models take more computation time than the static models. This is because of the construction of the optimal policy tables in the objective functions. In both static and dynamic OD-based models, the strategies based on finite differences (ODB-STD and ODB-DPD) require longer computation times than their counterparts based on bid prices (ODB-STB and ODB-DPB).

In Figures 8 and 9, we report our results for large-size networks with 1,000 and 5,000 OD-pairs, respectively. As we noted for the moderate-size networks, the strategies based on finite differences (FDD, ODB-STD, ODB-DPD) and leg-based decomposition

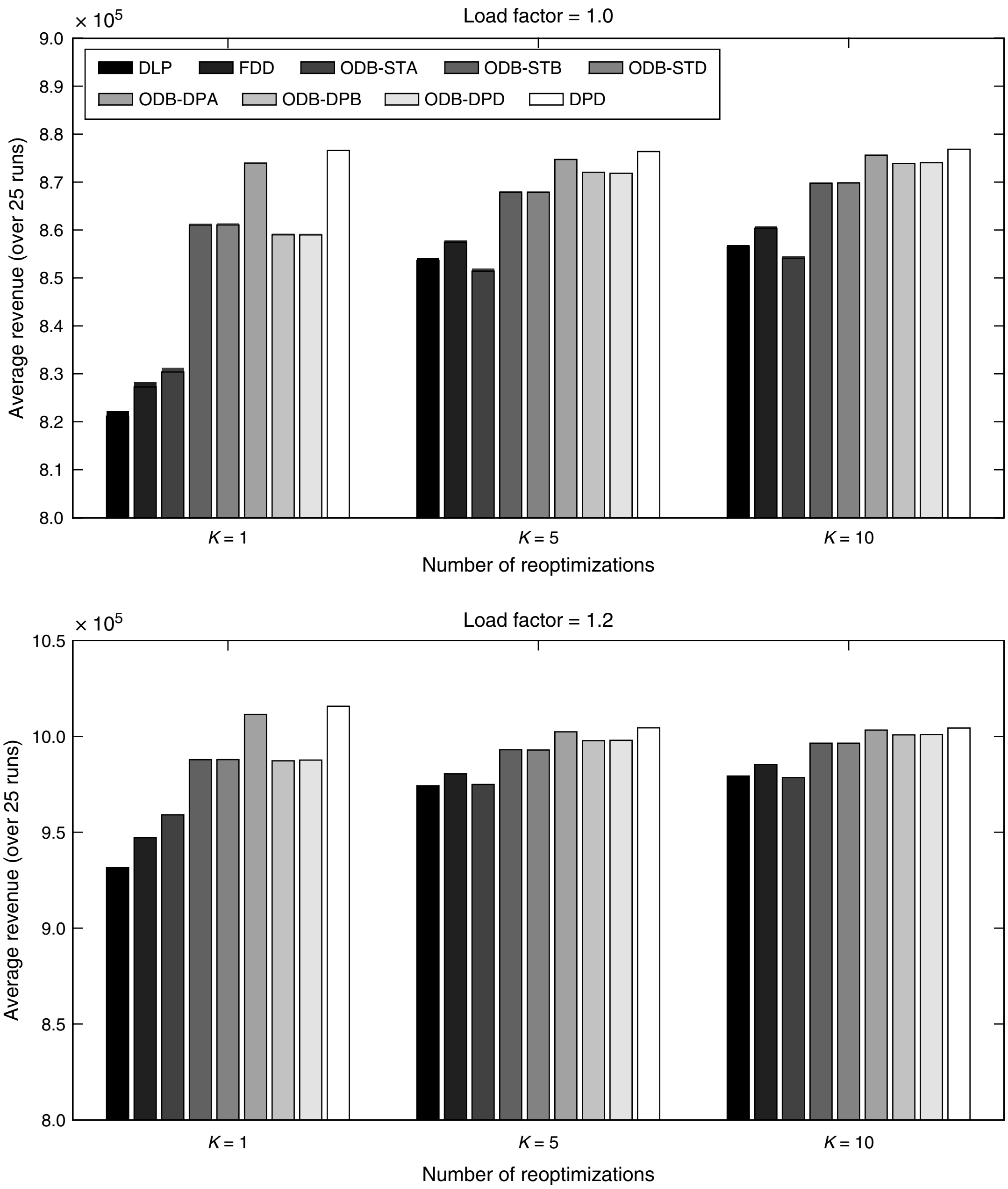
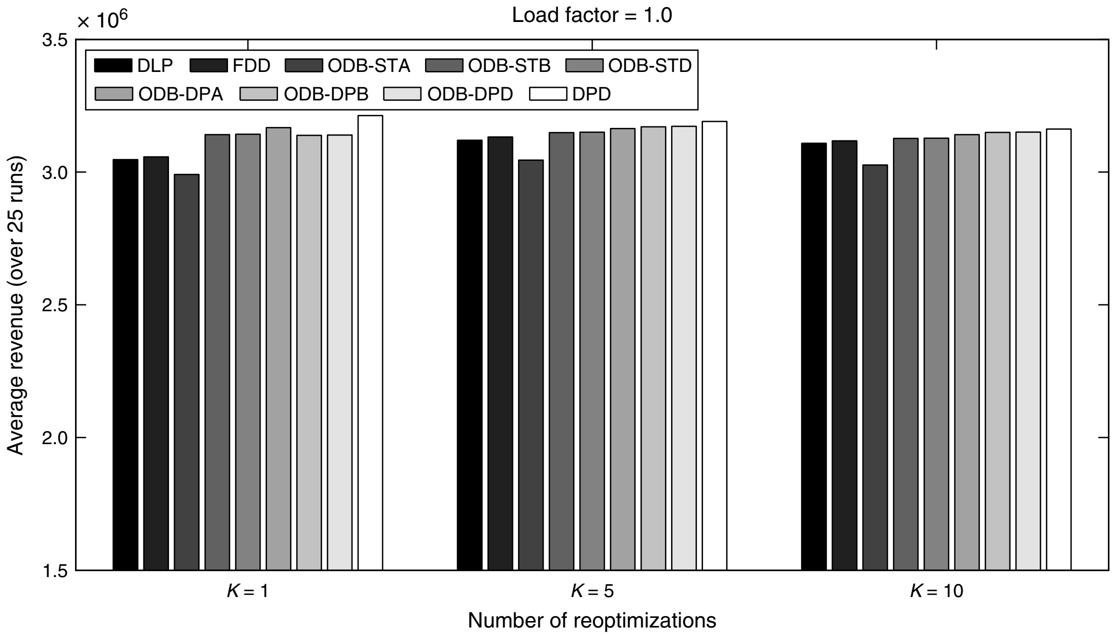


Figure 5 **The Average Revenues Obtained by All Strategies for Varying Load-Factor (ρ) and Reoptimization (K) Parameters (Network Size = 100 OD-Pairs)**

strategy (DPD) take an excessively long time, and their computational efforts depend exclusively on the network size. In addition, their performances are not significantly better than other strategies, especially the ones that are based on bid prices. Therefore, we do not consider the results with the finite-difference methods and DPD for the large-size networks.

Both Figures 8 and 9 show that when reoptimizations are carried out, the OD-based strategies based on bid prices, ODB-STB and ODB-DPB, are performing better than their allocation-based counterparts, ODB-STA and ODB-DPA, respectively. The main reason behind the performance decrease of allocation-based methods is their lack of consideration for the cross effects of capacity sharing. In other words, preallocating the capacities to each OD-pair independent of the other pairs undermines the proper control of networkwide inventories. The importance of capacity sharing is even more evident as the network size becomes larger. When a flight leg is shared among many OD-pairs, this causes the fragmentation of the flight-leg capacity, and hence, many seats remain unsold to some itineraries even if there is excess demand



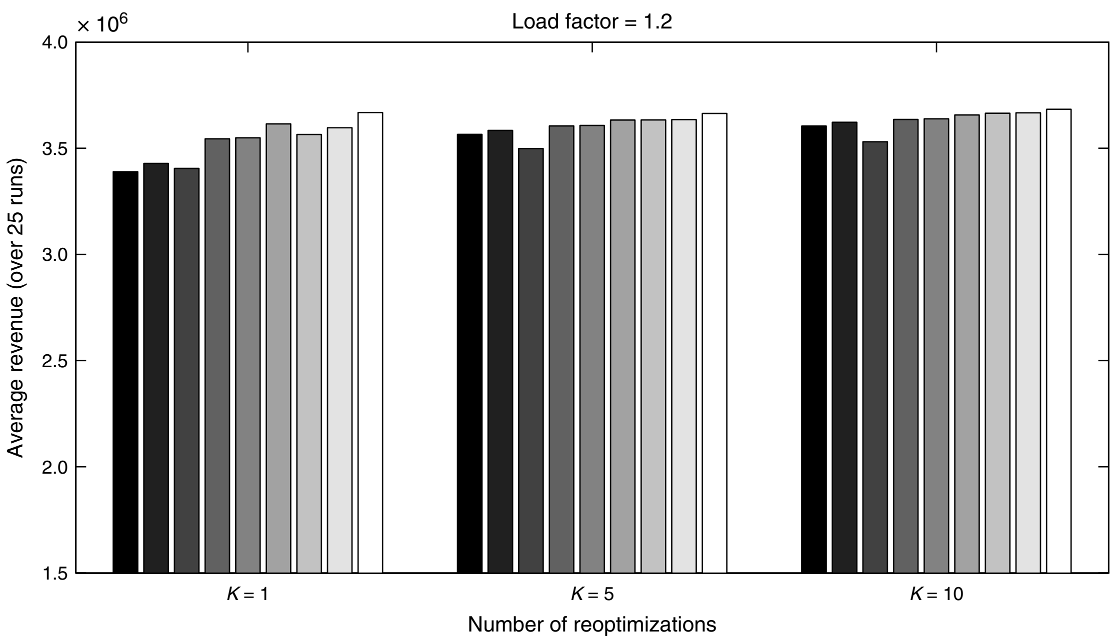


Figure 6 **The Average Revenues Obtained by All Strategies for Varying Load-Factor (ρ) and Reoptimization (K) Parameters (Network Size = 500 OD-Pairs)**

from other OD-pairs. We note that a similar concern has already been raised in the literature, particularly, by Curry (1990) and Williamson (1992). However, although allocation-based, the dynamic strategy ODB-DPA is still relatively competitive for the network with 1,000 OD-pairs (Figure 8). Among those strategies based on bid prices, ODB-DPB obtains the highest average revenue. The OD-based static model, however, does not perform equally well, and it is comparable to the famous DLP method. This difference between ODB-STB and the other strategies is in accordance with the observation of Williamson (1992), who also notes that the deterministic methods based on average demand values perform better than those methods based on probabilistic demand information. As de Boer et al. (2002) also observe, the static methods using probabilistic information seem to suffer more from ignoring the nesting of the allocations than the deterministic methods. Figures 8 and 9 clearly demonstrate that the DLP method is very successful as the size of the network becomes larger and the reoptimizations are carried out.

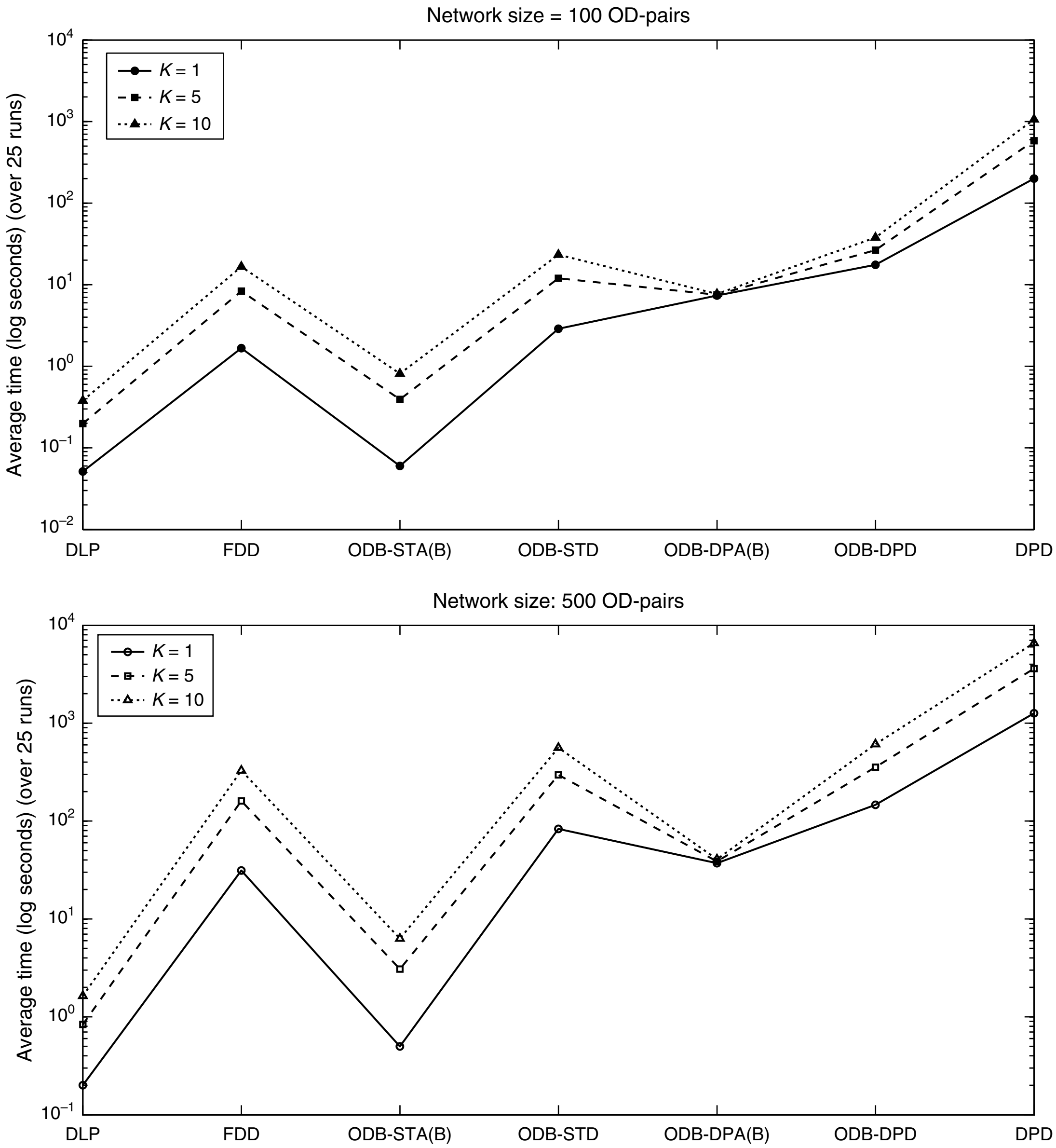


Figure 7 **The Average Computation Times of All Strategies for Moderate-Size Networks** ($\rho = 1.0, K = 1$)

We also compare the average computation times of the benchmarking strategies used for solving large-size networks. Although their performances are quite good, the dynamic OD-based strategies take a longer time than DLP. Unlike the moderate-size problems, we observe that the construction and the solution time of the linear programming problem obtained by the reformulation of (10)–(12) takes some time when the network size increases. Therefore, we see a slight increase in the computation times of the dynamic OD-based strategies when reoptimizations are carried out.

Overall, our results demonstrate that if the network size is moderate and reoptimizations are not possible, then the allocation-based strategy, ODB-DPA, is a proper choice to obtain high revenues. When reoptimization can be applied and the network size becomes larger, the strategies based on bid prices take over, and ODB-DPB becomes the most successful one among them. However, this performance gain comes at the expense of increased computation time. When the network becomes quite large and congested (high load factor), the performance of DLP is noteworthy. This well-known strategy obtains high revenues with a minor computational effort.

At the end of §3.1, we mentioned that the optimal objective function values of the OD-based dynamic model (10)–(12) and the deterministic linear program give lower and upper bounds for the complete dynamic model, respectively. We observe that the percentage differences between these two values are less than 5% for the first three network sizes. However, this difference goes slightly above 8% for

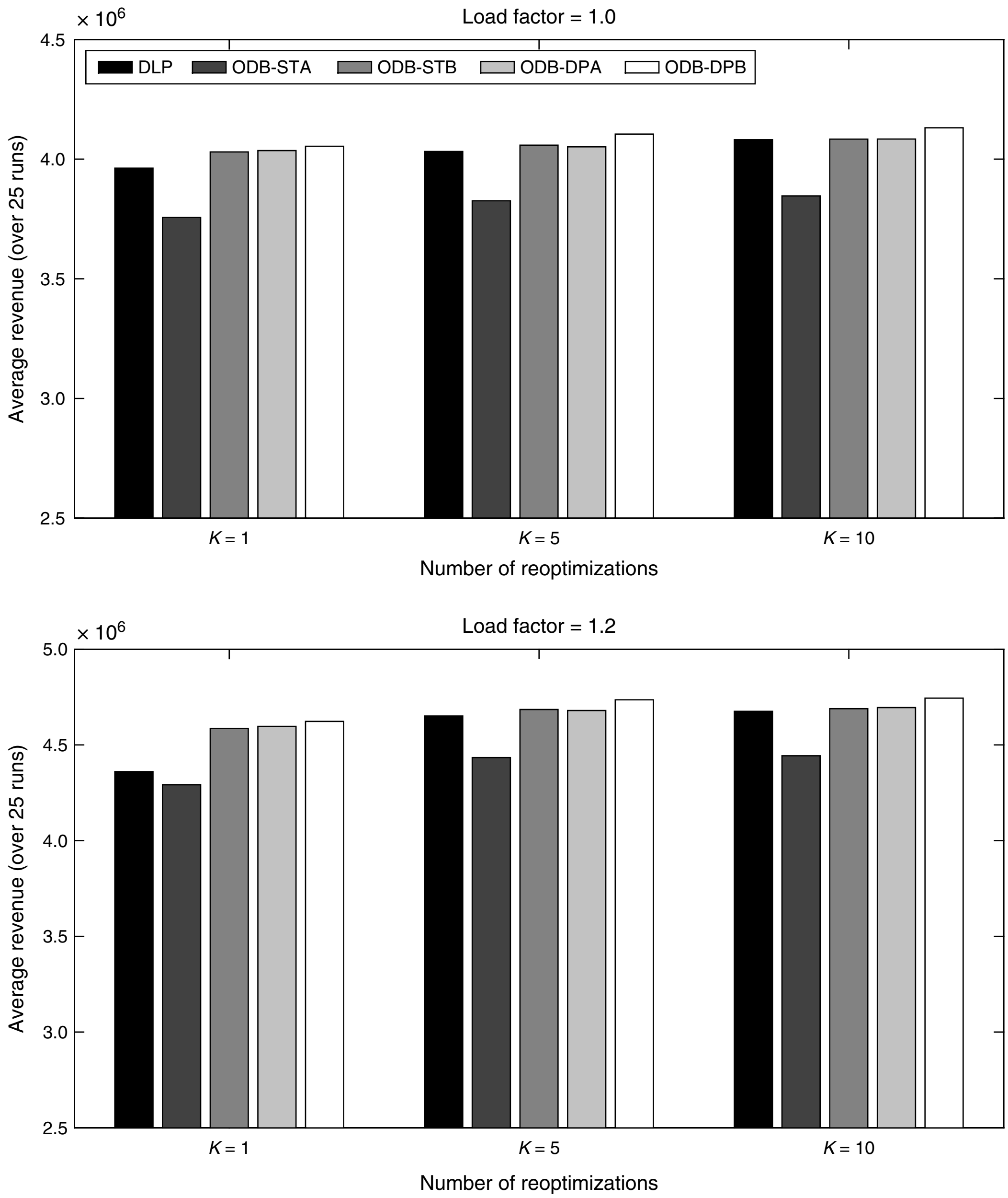


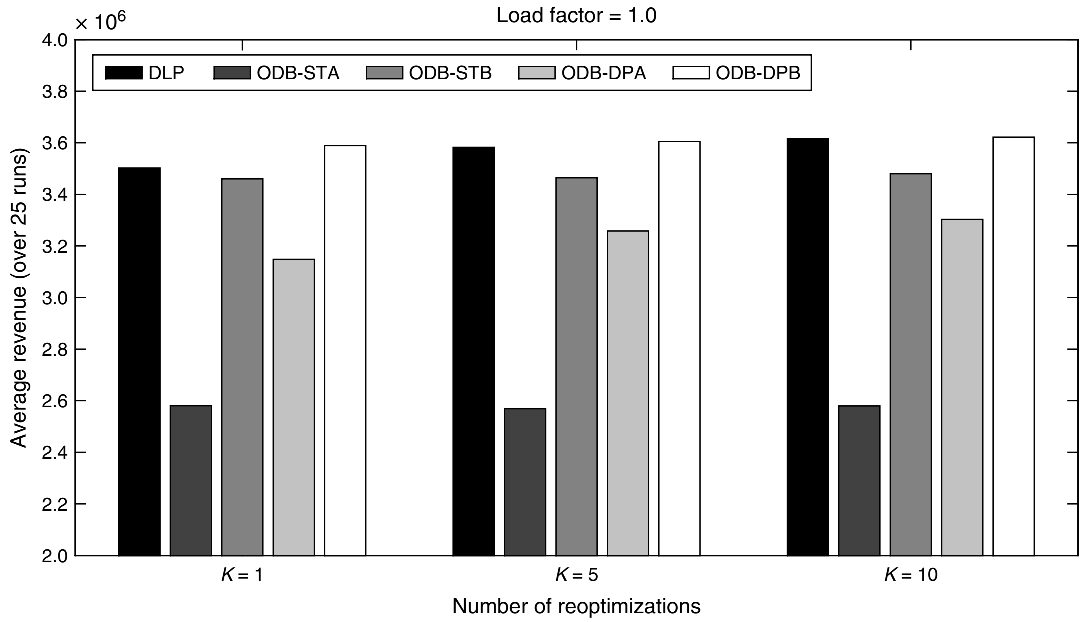
Figure 8 **The Average Revenues Obtained by All Strategies for Varying Load-Factor (ρ) and Reoptimization (K) Parameters (Network Size = 1,000 OD-Pairs)**

the largest network. These figures are even better as the load factor increases. These observations indicate that the OD-based dynamic model could be a useful tool for decision makers also for estimating the loss in revenue due to not being able to solve the complete dynamic model.

Our last discussion in this section is about the gaps between the objective function values of the linear programming relaxation optimal solution and the rounded-down integer-feasible solution for both static and dynamic OD-based models. Our results demonstrate that, as expected, the average gap increases slightly when the network size increases. However, one observation is crucial: the average gap increases almost linearly with a quite small slope. In all of the experiments, the average gaps for the dynamic models are slightly better than those of the static models. When it comes to the effect of the load factor, we observe that the difference between static and dynamic models in terms of the average gap decreases marginally as the load factor increases.

5. Conclusions and Future Research

We have discussed that the proposed generic framework is quite promising for extending various single-leg models to a network setting. We have illustrated



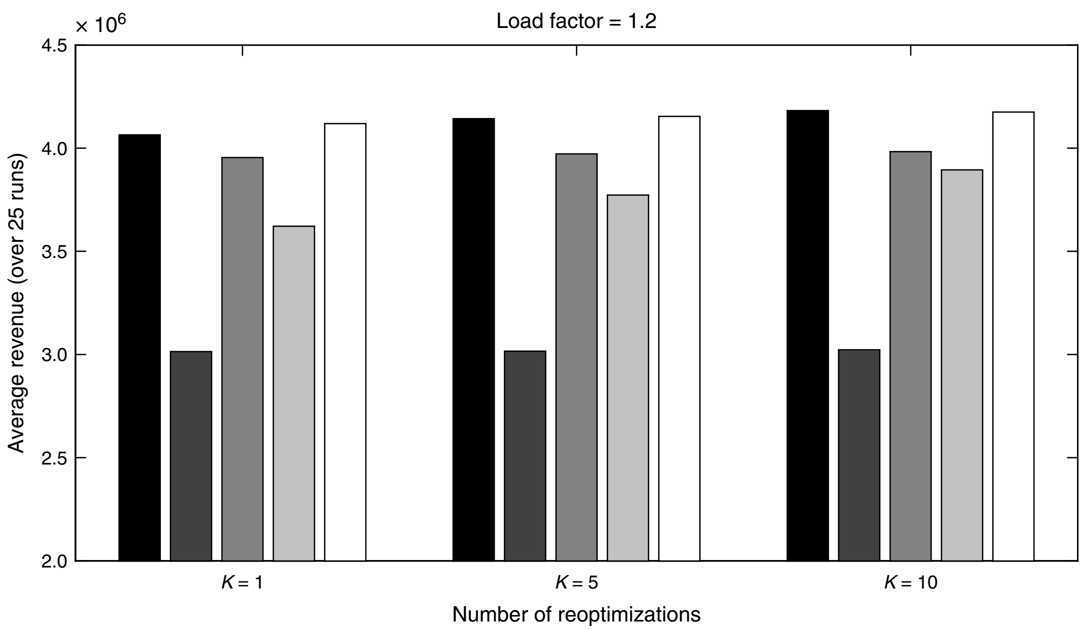


Figure 9 **The Average Revenues Obtained by All Strategies for Varying Load-Factor (ρ) and Reoptimization (K) Parameters (Network Size = 5,000 OD-Pairs)**

this observation through two illustrative examples, one of which led to a new model that was not considered before in the literature. We have also discussed that several well-known models from the literature can also be modeled in our framework, and consequently, one may save some computational effort by solving them with the proposed two-stage approach.

Our computational study on a set of real-life instances demonstrates that when the network size is moderate, OD-based strategies using allocations, in particular the dynamic model, perform quite well. However, as the network size becomes larger, the OD-based strategies relying on bid prices take the lead. Overall, the results obtained by the proposed OD-based dynamic models are very encouraging. By solving the dynamic model, one obtains both the allocations and the bid prices. Therefore, moderate or large, different network problems with varying sizes can be solved effectively by the proposed OD-based dynamic model. Furthermore, the optimal objective function value of the OD-based dynamic model gives a lower bound for the complete dynamic model that suffers from the curse of dimensionality. When used with the upper bound obtained from the well-known deterministic linear program, the difference between these upper and lower bounds can help the decision makers to estimate their loss by committing to a heuristic rather than solving the complete dynamic model. A possible disadvantage of using the proposed OD-based decomposition, however, may be undermining the impact of capacity sharing. Because the capacity allocated to each OD-pair is independent of the other pairs, the control of the networkwide inventories may be poor. This performance decrease can be more significant as the network becomes large and congested.

We have shown that the generic mathematical programming model is $\mathcal{NP}$-hard. However, our numerical experiments have indicated that solving the linear programming relaxation and rounding down its optimal solution gives an integer solution with an objective function value close to the optimal value. One may invest more effort in identifying an even better integer solution, but we believe solving the integer problem to optimality is not the major issue. As also observed by other researchers, it seems much more important to come up with a new method that considers the cross effects of capacity sharing among the OD-pairs when the network size becomes very large.

As we mentioned in §3.3, the recent studies in single-leg airline revenue management concentrate on customer choice models as well as on those models that consider inaccuracies with the stochastic information. The optimal policies proposed by some of these single-leg models lead to objective functions that are discrete concave in terms of the capacities. Therefore, these models can also be extended to a network setting with the proposed construction here. An important issue in revenue management is overbooking. Again, following our construction here, the overbooking problem over a network of flights can also be solved in two stages; for instance, Aydın et al. (2013) gives approximate overbooking models for single-leg models that satisfy our assumptions. However, if one partitions the capacities and determines the overbooking limits for each OD-pair independently, then these limits would accumulate on flight legs that are used by many itineraries. This would then not only overshoot the overbooking cost but also makes it significantly more difficult to decide which itinerary's customers should be denied boarding. Using the bid prices of the overall problem may be a remedy to avoid the partitioning of the capacities, but the challenge of evaluating the overbooking cost correctly remains.

Acknowledgments

The authors appreciate the comments offered by the two referees. Their comments certainly helped us improve this work considerably. They also thank Hüseyin Topaloğlu from Cornell University and Nilay Noyan from Sabancı University for their comments on an earlier version of this work. The authors also thank Nurşen Aydın from Sabancı University for her invaluable help with the computational study. The first author is supported by The Scientific and Technological Research Council of Turkey (TÜBİTAK) with research fellowship 2219.

References

Adelman D (2007) Dynamic bid prices in revenue management. *Oper. Res.* 55(4):647–661.

Aydın N, Birbil Şİ, Frenk JBG, Noyan N (2013) Single-leg airline revenue management with overbooking. *Transportation Sci.* 47(4):560–583.

Ball MO, Queyranne M (2009) Toward robust revenue management: Competitive analysis of online booking. *Oper. Res.* 57(4):950–963.

Belobaba PP (1987) Air travel demand and airline seat inventory management. Ph.D. thesis, Flight Transportation Laboratory, Massachusetts Institute of Technology, Cambridge, MA.

Bertsimas D, de Boer S (2005) Simulation-based booking limits for airline revenue management. *Oper. Res.* 53(1):90–106.

Bertsimas D, Popescu I (2003) Revenue management in dynamic network environment. *Transportation Sci.* 37(3):257–277.

Birbil Şİ, Frenk JBG, Gromicho JAS, Zhang S (2009) The role of robust optimization in single-leg airline revenue management. *Management Sci.* 55(1):148–163.

Brumelle SL, McGill JI (1993) Airline seat allocation with multiple nested fare classes. *Oper. Res.* 41(1):127–137.

Cooper WL, Homem-de-Mello T (2007) Some decomposition methods for revenue management. *Transportation Sci.* 41(3):332–353.

Curry RE (1990) Optimal airline seat allocation with fare classes nested by origins and destinations. *Transportation Sci.* 24(3):193–204.

de Boer SV, Freling R, Piersma N (2002) Stochastic programming for multiple-leg network revenue management. *Eur. J. Oper. Res.* 137(1):72–92.

Fourer R (1992) A simplex algorithm for piecewise-linear programming III: Computational analysis and applications. *Math. Programming* 53(1–3):213–235.

Glover F, Glover R, Lorenzo J, McMillan C (1982) The passenger-mix problem in the scheduled airlines. *Interfaces* 12(3):73–80.

Huh WT, Rusmevichientong P (2006) Adaptive capacity allocation with censored demand data: Application of concave umbrella functions. Technical report, School of Operations Research and Information Engineering, Cornell University, Ithaca, NY.

Kunnumkal S, Topaloglu H (2009) A stochastic approximation method for the single-leg revenue management problem with discrete demand distributions. *Math. Methods Oper. Res.* 70(3):477–504.

Kunnumkal S, Topaloglu H (2011) A stochastic approximation algorithm to compute bid prices for joint capacity allocation and overbooking over an airline network. *Naval Res. Logist.* 58(4):323–343.

Lan Y, Gao H, Ball MO, Karaesmen I (2008) Revenue management with limited demand information. *Management Sci.* 54(9):1594–1609.

Lautenbacher CJ, Stidham S Jr (1999) The underlying Markov decision process in the single-leg airline yield-management problem. *Transportation Sci.* 33(2):136–146.

Lee TC, Hersh M (1993) A model for dynamic airline seat inventory control with multiple seat bookings. *Transportation Sci.* 27(3):252–265.

Liang Y (1999) Solution to the continuous time dynamic yield management model. *Transportation Sci.* 33(1):117–123.

Littlewood K (1972) Forecasting and control of passengers. *AGIFORS Sympos. Proc., Nathanya, Israel*, Vol. 12, 95–117.

Perakis G, Roels G (2010) Robust controls for network revenue management. *Manufacturing Service Oper. Management* 12(1): 56–76.

Simpson RW (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.

Smith BC, Leimkuhler JF, Darrow RM (1992) Yield management at American Airlines. *Interfaces* 22(1):8–31.

Srinivasan A (1999) Improved approximation guarantees for packing and covering integer programs. *SIAM J. Comput.* 29(2): 648–670.

Talluri K (2008) On bounds for network revenue management. Accessed January 2013, http://ssrn.com/abstract=1107171.

Talluri K, van Ryzin G (1998) An analysis of bid-price controls for network revenue management. *Management Sci.* 44(11, Part 1):1577–1593.

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

Talluri KT, van Ryzin GJ (2005) *The Theory and Practice of Revenue Management* (Springer, New York).

Tong C, Topaloglu H (2014) On the approximate linear programming approach for network revenue management problems. *INFORMS J. Comput.* 26(1):121–134.

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

Topaloglu H (2009) Using Lagrangian relaxation to compute capacity-dependent bid prices in network revenue management. *Oper. Res.* 57(3):637–649.

van Ryzin G, McGill J (2000) Revenue management without forecasting or optimization: An adaptive algorithm for determining airline seat protection levels. *Management Sci.* 46(6):760–775.

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

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

Williamson EL (1992) Airline network seat control. Ph.D. thesis, Operations Research Center, Massachusetts Institute of Technology, Cambridge, MA.

Wollmer RD (1986) A hub-and-spoke seat management model. Technical report, Douglas Aircraft Company, McDonnell Douglas Corporation, Long Beach, CA.

Zhang D (2011) An improved dynamic programming decomposition approach for network revenue management. *Manufacturing Service Oper. Management* 13(1):35–52.