We investigate dynamic policies for allocating scarce inventory to stochastic demand for multiple fare classes, in a network environment so as to maximize total expected revenues. Typical applications include sequential reservations for an airline network, hotel, or car rental service. We propose and analyze a new algorithm based on approximate dynamic programming, both theoretically and computationally. This algorithm uses adaptive, nonadditive bid prices from a linear programming relaxation. We provide computational results that give insight into the performance of the new algorithm and the widely used bid-price control, for several networks and demand scenarios. We extend the proposed algorithm to handle cancellations and no-shows by incorporating oversales decisions in the underlying linear programming formulation. We report encouraging computational results that show that the new algorithm leads to higher revenues and more robust performance than bid-price control.

Introduction

Capacity constrained service industries, such as transportation, tourism, entertainment, media, and internet providers are constantly faced with the problem of intelligently allocating their limited, perishable inventories to demand from different market segments, with the objective of maximizing total revenues. Revenue management is concerned with the theory and practice underlying this type of problem. Following airline deregulation, revenue management techniques have had an important impact on the development of the industry, providing up to 4%–10% increases in company revenues (Fuchs 1987). For example, in 1997, American Airlines collected one billion dollars by implementing revenue management, representing most of the company’s profit (Cook 1998).

Optimization techniques have been essential in the development of revenue management tools, particularly for seat allocation models. In this research, we are interested in investigating the design of dynamic policies for allocating inventory to correlated, stochastic demand for multiple classes, in a network environment. Specifically, we design a decision support tool, based on stochastic and dynamic optimization techniques, that at each point in time accepts or rejects a reservation request, based on the currently available inventory, past sales and future potential demand, so as to maximize total expected revenues.

Problem Definition

The main problem we address in this paper is as follows. We are given an airline (hotel, car rental) network composed of $l$ legs (pairs of consecutive days for hotels, car rentals), which are used to serve a total of $m$ demand classes. The initial inventory is given by a vector $N = (N_1, \ldots, N_l)$ of leg capacities. The network is described by a $l \times m$ matrix $A$ and a $m$-vector $R = (R_1, \ldots, R_m)$; $R_j$ is the fare category of class $j$, which utilizes $a_{ij}$ units of resource (leg) $i$. In this way,
a demand class \( j \) is defined by its itinerary \( A^j \) (a column of matrix \( A \)) and its fare category \( R_j \). In an airline network without group discounts, \( A \) is a 0–1 matrix which may contain repeated columns for each fare class on a given itinerary. To account for special group fares, integer multiples of the itinerary-incidence vector are allowed.

For example, consider a very simple network corresponding to a weekend in a hotel: There are three nodes \( 	ext{Fri, Sat, Sun} \) and \( l = 2 \) legs (1) \( 	ext{Fri–Sat} \) and (2) \( 	ext{Sat–Sun} \) with total capacities \( \mathbf{N} = (N_1, N_2) \). Suppose there is demand for all types of stays, i.e., “itineraries”: (1) \( 	ext{Fri–Sat} \), (2) \( 	ext{Sat–Sun} \), and (3) \( 	ext{Fri–Sat–Sun} \), with two (high and low) fare classes for each type. Moreover, suppose there are discounts for groups of size \( k_j = 10 \) for (1) \( 	ext{Fri–Sat} \) night stays, at a rate of \( R_1^{(10)} \) per group, that is a total of \( m = 7 \) classes. The leg-class incidence matrix, together with the corresponding fare structure \( \mathbf{R} \), is given by:

\[
\begin{pmatrix}
R_1^h & R_1^l & R_2^h & R_2^l & R_3^h & R_3^l & R_1^{(10)} \\
1 & 1 & 0 & 0 & 1 & 1 & 10 \\
0 & 0 & 1 & 1 & 1 & 1 & 0
\end{pmatrix}
\]

We assume a finite booking horizon of length \( T \), with the time line sufficiently discretized so as to allow at most one request (reservation or cancellation) per time period almost surely (a.s.). Time is counted backwards: Time \( t = T \) is the beginning of the booking horizon and time \( t = 0 \) is the end of the reservation period, where the no-shows are being counted. Customers who do not consume their reservations get full, partial, or no refund, depending on their fare class. If at the end of the horizon, the inventory is oversold, at time \( t = -1 \) redistribution decisions are being made. These can be class upgrades or customer bumping, in which case companies pay overbooking penalties. For the case of hotels and car rentals, an infinite time horizon could be more appropriate; however, one can decompose the problem in fixed length time periods (one month, one year, etc.).

The demand (to come) process at time \( t \) is denoted by \( D^t \), and \( \overline{D}^t \) represents the corresponding random vector of cumulative demands. That is, \( D_j^t \) is a random variable representing the number of class \( j \) requests to come from time \( t \) until departure (order does not matter). Usually, we have partial information about the demand process, which might consist of the expected demand to come \( D^t = E[D^t] \), and possibly other type of information available from forecasting tools, such as cancellation or no-show probabilities.

The state of the system \( S \) is given by the time \( t \) (\( t \) periods to departure) and the sales-to-date record \( s = (s_1, \ldots, s_m) \) for each demand class. If cancellations and no-shows are not allowed, then it is sufficient to define the state based on the remaining inventory \( n = (n_1, \ldots, n_1) \). A common practice is to use the latter model and account for cancellations and no shows by incorporating a virtual increase, called overbooking pads, in the initial capacity definition.

The general stochastic, dynamic inventory control problem for network revenue management (NRM) can thus be stated as follows: At time \( t \), and given that the state of the network is \( S \), should we accept or reject a new class \( j \) request? The overall objective is to maximize total expected revenues.

The decision to accept or reject determines an admission control policy, which is in general a function of the current network configuration (\( s \) or \( n \)), the time-to-go \( t \), the currently requested fare \( R_j \), as well as partial information from demand forecasts. If we accept the request, the new state becomes \( (s + k_j \cdot e_j, t - 1) \), where \( k_j \) is the size of the class \( j \) group request, and \( e_j \) is the \( j \)th unit vector; when cancellations are not allowed, the state of the network \( (n, t) \) becomes simply \( (n - A', t) \). If the request is rejected, only the time component of the state vector is changed to \( t - 1 \).

In reality, the control policy has an impact on the demand process, because customer choice may depend on the opportunity set. Capturing and quantifying this type of feedback phenomenon is a subtle task that goes beyond the scope of this work. We will thus make the simplifying assumption that the demand process is independent of the control policy.

**Notation**

We will use the following notation throughout the paper. Vectors will be denoted in bold, and random variables and processes in caligraphic style.
Time:
\[ T = \text{length of time horizon (number of time periods)}; \]
\[ t = \text{time periods left until departure (count-down)}. \]

Network:
\[ l = \text{number of legs in the network}; \]
\[ m = \text{number of classes (itineraries with fare categories)}; \]
\[ \mathbf{N} = \text{total initial network capacity (l-vector)}; \]
\[ \mathbf{A} = \text{leg-class incidence (l \times m)-matrix}; \]
\[ a_{ij} = \text{quantity of resource i utilized by bundle j}. \]

Sales and inventory:
\[ n = \text{remaining inventory (l-vector)}; \]
\[ s = \text{sales to date vector (m-vector)}; \]
\[ s_j^t = \text{the number of itineraries sold to class j until time } t; \]
\[ s_j^t = \text{the number of class j itineraries overbooked at departure}. \]

Fares, refunds, and penalties:
\[ R_j = \text{revenue collected for one class j sold}; \]
\[ R_j = \text{the refund per class j cancellation}; \]
\[ R_j^* = \text{the refund per class j no-show}; \]
\[ C_j = \text{the overbooking penalty per demand class j}. \]

Demand:
\[ p_j^t = \text{probability of request for class j at time t}; \]
\[ p_j^t = \text{the probability of a class j cancellation occurring at time t}; \]
\[ p_0 = \text{the probability of no request (reservation or cancellation) at time t}; \]
\[ p_j^t = \text{the probability that a class j reservation will not show up}; \]
\[ \mathbf{D} = \text{demand (to come) process (m-dimensional)}; \]
\[ \overline{\mathbf{D}} = \text{aggregate demand (to come) distribution (m-dimensional)}; \]
\[ \mathbf{D} = E[\overline{\mathbf{D}}] = \text{expected aggregate demand to come (m-dimensional)}; \]
\[ \mathbf{C} = \text{the cancellation process, adapted to the sales history (not a decision)}; \]
\[ N = \text{the no-show distribution (adapted to final sales, adjusted for cancellations)}; \]
\[ \mathbf{c} = \text{the opportunity cost of each itinerary is estimated as the sum of the shadow prices of the incident legs, obtained from a linear programming formulation of the problem (see Formulation (2) in §2.2). However, there are two obvious drawbacks to additive bid prices:} \]

(a) They are not well defined if there are multiple dual solutions.

(b) They are restrictive in their way of taking into account bundles by their predefined additive structure. In particular, they do not account for changes of a dual basis in response to accepting large-group and multileg itinerary requests.

The contributions of this paper are as follows:

(1) We propose an efficient control policy, that is well defined if there are multiple dual solutions, and does not have an additive structure. The proposed control policy, which we call certainty equivalent control (CEC), belongs in the class of approximate dynamic programming mechanisms (see Bertsekas and Tsitsiklis 1998), in which the cost-to-go function is approximated by the value of a linear programming (LP) relaxation. We remark that this LP is equivalent to a network flow problem in the case of origin-destination (OD) demands or linear networks (where nodes can be ordered so that the arcs are pairs of consecutive nodes \((i, i+1), i = 1, \ldots, l\)).

(2) We provide structural properties that compare the behavior of the proposed CEC policy with the additive bid-price approach. These results offer insight into the behavior of both methods.

(3) We propose several algorithmic improvements of the CEC policy based on approximate dynamic programming.

(4) We provide computational results that give insight into the performance of these algorithms and several variations, for different networks and demand scenarios. We observe that the CEC algorithm performs very well in practice, giving results that are very close to optimum. For high load factors, we observe an average 5%–10% improvement over existing policies (additive bid pricing). We describe and
simulate extensions of this algorithm that result in significantly higher improvements (up to 20%). Interestingly, the CEC policy appears to be significantly more robust to noise and bias in the demand forecast.

(5) We extend these algorithms to handle cancellations and no-shows by incorporating overbooking control in the underlying mathematical programming formulation. This extension preserves several structural properties. We report computational results that show that the proposed algorithm improves upon the performance of the bid-price control policy.

Structure
The remainder of the paper is organized as follows: In the next section, we present an overview of the literature. Section 2 describes several (dynamic, stochastic, linear, and network flow) models and formulations for the NRM problem and evaluates the relationships between them. Section 3 presents efficient algorithms for the NRM problem based on ideas from approximate dynamic programming. In §4, we present structural properties for the proposed CEC policy and contrast them with additive bid pricing. In §5, we extend our model and algorithms to handle cancellations, no-shows, and overbooking. Finally, in §6, we present computational results. The last section summarizes our conclusions.

1. Literature Review and Positioning
The NRM problem can be viewed as a particular instance of the general class of perishable asset revenue management problems (PARM). Initial developments in static single leg revenue management are due to Littlewood (1972), followed by Simpson (1989) and Belobaba (1987), who proposed a suboptimal policy for computing protection levels based on expected marginal seat revenues (EMSR). Curry (1989), Wollmer (1992), and Brumelle and McGill (1993), derive the optimal solution for the single leg static model. Robinson (1995) proposes an extension that handles nonmonotonic fare classes.

A characterization of the optimal dynamic policy based on a threshold time property is due to Diamond and Stone (1991). An analogous discrete time solution is provided by Lee and Hersh (1993), who also provide a method for estimating arrival rates. Subramanian et al. (1999) propose a dynamic programming model that handles cancellations and overbooking, by analogy to a problem in the optimal control of admission in a queueing system.

A natural, but much harder question is to determine which of the single leg results can be extended to network (multiproduct) settings, and how. A major conceptual advance in the study and practice of network revenue management was introduced by bid-price control. These are additive, leg-based shadow prices used to approximate the opportunity cost of itinerary capacity. The concept was proposed by Simpson (1989), and further analyzed by Williamson (1992) in extensive simulation studies. A deficiency of such mathematical programming based models is that they do not account for nesting of the fare classes.

To overcome this problem, Curry (1992) proposes a virtual nesting method. In all cases, however, the allocation of capacity to itinerary demand is decided by a one-time, static rule (one fixed set of bid prices).

The most realistic and relevant, yet least investigated model for NRM is the dynamic network model. Talluri and van Ryzin (1998) study a dynamic network model using bid-price control mechanisms and argue why bid-price policies are not optimal in general. They provide an asymptotic regime when certain bid-price controls, based on a probabilistic programming formulation of the problem, are asymptotically optimal. In the context of hotels, Bitran and Mondschein (1995) propose a dynamic policy that extends to multiple-night stays, but do not give any further analysis.

Chen et al. (2000) formulate the problem as a Markov decision problem, and use linear programming and regression splines to approximate the value function. Gunther et al. (2000) introduce a new method to compute bid price for single hub airline networks. Both studies report encouraging simulation results. None of these take into account cancellations.

For further pointers to related literature, we refer the interested reader to the survey of McGill and...
van Ryzin (1999), and for a schematic summary the Ph.D. thesis of Popescu (1999).

**Relative Positioning.** Our approach can be viewed as extending the single-leg models investigated by Lee and Hersh (1993) and Bitran and Mondschein (1995), who actually propose a similar LP-based heuristic for the multiple-night booking problem in a hotel, but without any further analysis. Our overbooking model is similar to the early single-leg DP formulation of Rothstein (1971, 1974). As far as static network models are concerned, our LP formulation is similar to the one proposed by Williamson (1992), but we handle multiple classes and group bookings. The network flow formulation we provide for linear networks is concerned, our LP formulation is similar to the one proposed by Williamson (1992), but we handle multiple classes and group bookings. The network flow formulation we provide for linear networks and origin-destination fare (ODF) demands, is the same as those proposed by Glover et al. (1982) for airlines and Chen (1998) for hotels.

### 2. General Models and Relationships

The problem of dynamic inventory control for NRM belongs in the class of finite horizon decision problems under uncertainty. In this section, we present several optimization models for addressing the NRM problem, and discuss various relationships between them. Again, we assume that the control policy does not feed back into the evolution of the demand process. We restrict to the case when cancellations and overbooking are not permitted; this case is discussed in detail in §5.

#### 2.1. The Dynamic Programming Model

The stochastic dynamic programming model provides the optimal policy for the NRM problem, by evaluating the whole tree of possibilities and making at each point in time the decision (to sell or not to sell) that would imply higher future expected revenues.

The states $S = (n, t)$ are defined by the current available capacity vector $n$ when the remaining time is $t$. The stochasticity is given by the demand process to come $D_t$ for the remaining $t$ periods. We define $DP(n, t)$ to be the maximum expected revenue to be collected from state $(n, t)$. Assuming independent demands, this can be computed via the Bellman equation as follows:

$$DP(n, t) = \max_{i} \left[ p_i D_P(n, t) + DP(n - A', t - 1) \right]$$

for all $n \leq N$, $t \leq T$, with the boundary conditions:

$$DP(n, 0) = 0, \quad \text{for } n \geq 0.$$  

Then, the optimal policy accepts a request if and only if the corresponding fare $R_i$ exceeds its current opportunity cost. The value function can thus be expressed as follows:

$$DP(n, t) = DP(n, t - 1) + \max_{i} \left[ R_i - OC_i(n, t) \right]$$

One can easily show that the value function is non-decreasing in $n$ and $t$. Moreover, for the single leg case without batch arrivals, the value function is concave and opportunity costs decrease with $t$ and $n$ (see Diamond and Stone 1991). Based on these properties, one can show that the optimal policy is characterized by threshold times. Threshold times are points in time during the booking horizon before which requests are rejected, and after which requests are accepted. In the general network case, however, this is not true, as we will show in an example in §4.1.

#### 2.2. The Integer and Linear Programming Model (IP, LP)

The most frequently utilized formulations for the NRM problem are static models. These are deterministic analogues of the stochastic dynamic problem,
that use only expected demand information, and are usually much simpler to solve.

Suppose that all the available demand information consists of (unbiased) forecasts of the expected aggregate demand to come \( D' = E[\mathcal{D}] \) over the remaining \( t \) periods. The integer programming model computes the optimal allocation \( y^* \) of available inventory \( n \) to the expected itinerary demand \( D' \), by maximizing total revenues subject to capacity and itinerary demand constraints. For all \( n \leq N \) and \( t \leq T \), we have

\[
\text{IP}(n, D') = \max_{y} \quad R' \cdot y \\
\text{s.t.} \quad A \cdot y \leq n \\
0 \leq y \leq D' \\
y \text{ integer.}
\]

The linear programming relaxation of this problem provides an efficient way to compute the best possible “fractional” allocation of inventory, and is defined by simply relaxing the integrality constraint:

\[
\text{LP}(n, D') = \max_{y} \quad R' \cdot y \\
\text{s.t.} \quad A \cdot y \leq n \\
0 \leq y \leq D'.
\] (2)

Clearly, we have that \( \text{IP}(n, D') \leq \text{LP}(n, D') \). This inequality can be strict, but there are particular instances when equality holds, one of which (linear networks) we describe in the next section. One can obtain further insight into the optimal LP-allocation by looking at the dual problem:

\[
\text{LP}(n, D') = \min_{v, u} \quad \nu \cdot n + (R' - \nu \cdot A) \cdot D' \\
\text{s.t.} \quad \nu' \cdot A + u \geq R' \\
u, v \geq 0.
\]

This formulation can be equivalently written as:

\[
\text{LP}(n, D') = \min_{\nu, u} \nu \cdot n + (R' - \nu \cdot A) \cdot D' \\
= \min_{k \in K} \nu_k \cdot n + u_k \cdot D',
\]

where \( K \) denotes the index set of extreme points \((v_k, u_k)\) of the dual polyhedron. Thus, the objective value of Model (2) is a piecewise linear, concave, and nondecreasing function of the expected demand to come \( D' \) and available capacity \( n \). We next relate the objective values of the dynamic and static formulations at any state \((n, t)\).

**Proposition 1.** \( \text{DP}(n, t) \leq \text{LP}(n, D') \).

For a full proof see Popescu (1999). The idea is that for each pathwise realization of demand, the revenue collected by the DP-policy along that path is upper bounded by the value of a “perfect hindsight” stochastic program. The perfect hindsight model, denoted \( \text{PI} \), determines the optimal allocation of inventory for each particular realization of demand, and then computes the expected reward over all scenarios:

\[
\text{DP}(n, t) \leq \text{PI}(n, \mathcal{D}) = E[\text{LP}(n, \mathcal{D})] \\
\leq \text{LP}(n, E[\mathcal{D}]) = \text{LP}(n, D').
\]

Here, \( \mathcal{D} \) is the cumulative demand corresponding to scenario \( \pi \). Since the LP value is concave in the demand vector, the second inequality follows from Jensen’s inequality. Furthermore, the two values converge as demand becomes very large (see Popescu 1999).

### 2.3. Origin-Destination RM and Linear Networks

Consider a particular case of the NRM problem, where demand is by ODF, as opposed to itinerary specific. This is by default the case for linear networks, where there is no question of routing, such as in hotel or car rental revenue management (where the nodes are days). In the case of airline revenue management this situation occurs when there are perfectly substitutable routes (same price, same travel time, and so on). In these cases, the Model (2) can be represented as a network flow problem. The advantage of this formulation is that it is very easy to solve in practice and reoptimization is very fast. For airlines, this model provides (for the company) optimal routing of path-indifferent customers.

The network representation is as follows: The nodes are the origins and destinations (days for hotels, airport, time)-pairs for airlines). There are multiple forward arcs \((o, d)\) representing the flow from origin \( o \) to destination \( d \) from fare-class \( f \). The capacity of each forward arc is the corresponding aggregate demand \( D'_d \). The revenue collected along arc \((o, d)\) is \( R'_d \). For each “leg” \((i, j)\) (flight leg, respectively, pair of consecutive days) there is a backward arc of the type \((j, i),\)
capacitated by \( n_{ij} \), the available inventory of leg \((i, j)\). The revenue collected along backward arcs is zero. This model has been proposed by Glover et al. (1982) in the context of airlines, and for hotel revenue management by Chen (1998), who proves that it reduces to solving a network flow problem.

Proposition 2 (Integrality of Solution). For ODF models with integer data, the optimal LP-solution is integral, that is \( \text{IP}(\mathbf{n}, \mathbf{D'}) = \text{LP}(\mathbf{n}, \mathbf{D'}) \).

3. Approximate Dynamic Programming Algorithms

In this section, we present several algorithms for the NRM problem. Most of these algorithms belong in the generic class of approximate dynamic programming methods (see Bertsekas and Tsitsiklis 1998), in which an approximate value to the exact value function is used in the Bellman equation.

Given a certain efficient mathematical programming formulation (MP) of the NRM problem, a generic approximate DP algorithm for the NRM problem has the following structure:

**Generic MP-Policy.** Identify an efficient formulation MP of the NRM problem.

At any current state \( \mathbf{S} = (\mathbf{n}, t) \),

1. For a class \( j \) request, compute an MP-based estimate of the opportunity cost \( OC^\text{MP}_j(S) \).
2. Sell to class \( j \) if and only if its fare \( R_j \) exceeds its opportunity cost estimate, i.e.,

   \[ R_j \geq OC^\text{MP}_j(S). \]

3. Go to Step 1 and ITERATE.

We will denote the expected value of this policy at an initial state \( \mathbf{S} \) as \( \Pi_{\text{MP}}(\mathbf{S}) \). The difference between various algorithms comes from the approximating MP and the MP-based opportunity cost measure, that is from Step 1.

3.1. Bid-Price Control

Bid-price control is a popular method in NRM, whereby the opportunity cost (shadow price, or bid price) of an itinerary is approximated by the sum of opportunity costs of the legs along that itinerary. First, opportunity cost estimates (bid prices) are determined for each leg in the network, usually as the leg-shadow prices \( v \) from the linear programming formulation (2). Then itinerary bid prices are computed additively, at each state \( \mathbf{S} = (\mathbf{n}, t) \), and for each class \( j \) request as

\[ \text{BP}_j(\mathbf{S}) = (v^5)' \cdot A^j. \]

Notice that bid prices depend on the choice of optimal dual variables \( v^5 \). This technique was initially proposed by Simpson (1989), then studied by Williamson (1992) in her Ph.D. thesis. Several probabilistic models have been investigated by Glover et al. (1982), Williamson (1992), and Talluri and van Ryzin (1998). It has been observed (Williamson 1992) that the LP model achieves a better performance, and is more efficient. For recent work on bid-price control see Talluri and van Ryzin (1998) and Gunther et al. (2000).

3.2. Certainty Equivalent Control

The main disadvantages that are apparent from the definition of leg-based additive bid prices is that (a) they are not uniquely defined (several sets of shadow prices may be optimal), and (b) they provide an additive approximation of the opportunity costs, which are not necessarily additive due to “bundle effects” (group or multileg itinerary requests may determine basis changes in the dual LP).

We provide a different approximation scheme for opportunity costs, based on certainty equivalent adaptive control. The idea is to approximate the value function of the dynamic program \( \text{DP}(\mathbf{n}, t) \) defined in Equation (1) by the value of the linear programming problem \( \text{LP}(\mathbf{n}, \mathbf{D'}) \) defined in §2.2. Thus, in Step 1 of the generic algorithm, a request for a given class \( j \) will be accepted if and only if its price \( R_j \) exceeds its current opportunity cost estimate given by:

\[ OC^\text{LP}_j(\mathbf{n}, t) = \text{LP}(\mathbf{n}, \mathbf{D'}^t - 1) - \text{LP}(\mathbf{n} - A^j, \mathbf{D'}^t - 1). \]

Because the opportunity cost estimate is calculated in terms of LP objectives, it is uniquely defined, in that its value will not depend on the choice of (dual) solution. Thereby, the first drawback of bid prices is resolved.

This is the certainty equivalent control (CEC) policy, and we denote its expected value for an initial state \( \mathbf{S} \) as \( \Pi_{\text{LP}}(\mathbf{S}) = \text{CEC}(\mathbf{S}) \). For more on certainty
equivalence, see Bertsekas (1995). A similar policy is proposed by Bitran and Mondschein (1995) in the context of hotel revenue management, but no analysis is provided. Notice that in the case of linear networks, it is actually desirable to use the equivalent network flow formulation as described in §2.3 instead of the usual LP model because reoptimization at each stage becomes a much simpler task.

More generally, one can use this technique with virtually any mathematical programming (MP) model that provides an approximation of the value function. The corresponding OC estimate for an itinerary \( j \) request at the current state \( S \) is defined as \( OC_j^{MP}(S) = MP(S_{\text{accept}}(j)) - MP(S_{\text{reject}}(j)) \), where \( S_{\text{accept}}(j) \) and \( S_{\text{reject}}(j) \) are the states corresponding to the accept, and respectively reject decision for the current request \( j \). For example, for the static LP model without cancellations and overbooking, if \( S = (n, t) \), then \( S_{\text{accept}}(j) = (n - \lambda^j, t - 1) \) and \( S_{\text{reject}}(j) = (n, t - 1) \).

3.3. Extensions

In this section, we describe several extensions that provide improvements in the efficiency or accuracy of the adaptive policies described above.

Rollout Policy. The expected value \( \Pi_H(S) \) of any heuristic \( H \) started in state \( S \) provides an approximation of the DP(S) value. Therefore, the expected heuristic values can in turn be used to provide estimates of opportunity costs, determined as follows:

\[
OC_j^{H}(n, t) = \Pi_H(n, t - 1) - \Pi_H(n - \lambda^j, t - 1).
\]

This opportunity cost estimation mechanism leads to a new approximate dynamic programming (ADP) heuristic, called the rollout of \( H \), and denoted \( R(H) \). This method is simply a form of policy iteration and is described in detail in Bertsekas and Tsitsiklis (1998). It has been observed in the dynamic programming literature that this procedure systematically improves heuristic performance (see also Bertsimas et al. 1999, Bertsekas et al. 1997, Bertsekas and Castanon 1998).

For practical purposes, we suggest using Monte Carlo simulation for evaluating the policy value \( \Pi_H \) for a subset of states, and then interpolating these in an online fashion. An interesting research idea is to investigate what types of preprocessing simulations would provide an insightful information database.

Simulations Using Monte Carlo Demand Estimation. One problem with the certainty equivalent policy is that it only considers expected demand information, and uses a deterministic approach to a highly stochastic problem. We propose a variation on the certainty equivalent policy that uses Monte Carlo demand estimation to capture demand variability. Suppose we have information that the cumulative demand to come \( \overline{\theta}_1^{-1} \) follows a certain distribution. We generate \( r \) samples from this distribution: \( \overline{\theta}_1^{-1}, \ldots, \overline{\theta}_r^{-1} \). In Step 1 of the generic algorithm, we calculate the opportunity cost estimate of itinerary \( j \) as a weighted average,

\[
OC_j^{MC}(n, t) = \sum_{i=1}^{r} \alpha_i OC_j^{LP}(n, \hat{D}_i^{-1}),
\]

where \( \alpha_i = P(\overline{\theta}_i^{-1} = \hat{D}_i^{-1} | \overline{\theta}_1^{-1}, \ldots, \overline{\theta}_r^{-1} \} \) and \( OC_j^{LP}(n, \hat{D}_i^{-1}) = LP(n, \hat{D}_i^{-1}) - LP(n - \lambda^j, \hat{D}_i^{-1}) \).

One difficulty with implementing this procedure is that we might not have enough information about the aggregate demand and/or it may be too expensive to compute the actual value of the conditional probabilities \( \alpha_i \). For this reason, we run a simplified version of this policy, that assigns the same weights to all the trials: \( OC_j^{MC}(n, t) = (1/r) \cdot \sum_{i=1}^{r} OC_j^{LP}(n, \hat{D}_i^{-1}) \).

4. Structural Properties

In this section, we derive several structural properties of the new approximate dynamic programming algorithm (CEC) and compare it with additive bid-price control algorithms (BPC) developed in the literature.

Given that the NRM problem requires a real-time response, it is desirable to use computationally inexpensive models to construct approximations of the opportunity cost. This motivates the choice for using the LP formulation described in §2.2. We are interested in a comparative structural assessment of the two LP-based policies described previously, BPC and CEC, and their corresponding opportunity cost approximations.

From an asymptotic point of view, it should be noted that the CEC policy is asymptotically optimal in the fluid scaling regime proposed by Talluri and van Ryzin (1998), whereby demand and capacities are simultaneously increased in a way that preserves
their relative values constant. They prove that in this regime, the additive bid-pricing policy converges to the optimum, as bid prices are being held fixed. By imitating their proof, one can show that the same property holds for the CEC policy, with deterministic prices, as OC estimates are being held fixed (see Popescu 1999).

The next result compares the opportunity cost approximations of a class $j$ request at any given state.

**Proposition 3.** In any state $(n, t)$, for any bid prices $BP_j(n, t), BP_j(n - A^j, t)$, the following inequalities hold:

$$BP_j(n, t) \leq OC_{sl}^j(n, t) \leq BP_j(n - A^j, t). \quad (4)$$

Inequalities are strict if accepting class $j$ must incur a change of basis in the LP dual.

**Proof.** Recall that we have defined:

$$OC_{sl}^j(n, t) = LP(n, D^{c-1}) - LP(n - A^j, D^{c-1})$$

$$= (v^{n, t})' \cdot n + (u^{n, t})' \cdot D^{c-1}$$

$$- (v^{n-A^j, t})' \cdot (n - A^j) - (u^{n-A^j, t})' \cdot D^{c-1},$$

where $(v^{n, t}, u^{n, t})$ and $(v^{n-A^j, t}, u^{n-A^j, t})$ are optimal dual solutions of $LP(n, D^{c-1})$ and $LP(n - A^j, D^{c-1})$, corresponding to the given bid prices: $BP_j(n, t) = (v^{n, t})' \cdot A^j$ and $BP_j(n - A^j, t) = (v^{n-A^j, t})' \cdot A^j$, respectively. Because both solutions are feasible for both programs, we obtain the following upper bounds by evaluating each LP at the optimal solution of the other:

$$LP(n, D^{c-1}) \leq (v^{n-A^j, t})' \cdot n + (u^{n-A^j, t})' \cdot D^{c-1},$$

$$LP(n - A^j, D^{c-1}) \leq (v^{n, t})' \cdot (n - A^j) + (u^{n, t})' \cdot D^{c-1}.$$

The upper bound in Equation (4) follows by applying the first of these inequalities in the OC formula, and the lower bound by the second inequality,

$$BP_j(n, t) = (v^{n, t})' \cdot A^j \leq OC_{sl}^j(n, t) \leq (v^{n-A^j, t})' \cdot A^j$$

$$= BP_j(n - A^j, t). \quad (5)$$

In case the two dual optimal solutions coincide, we obtain equality throughout. □

In general, if at a given state the CEC policy accepts a class $j$ request, then at the same state, the bid-pricing policy will also accept, but not vice versa. This is because the following situation may occur: $BP_j(n, t) \leq R_j < OC_{sl}^j(n, t)$, (see §4.1).

Table 1 provides a comparative characterization of the bid-pricing policy (BPC) versus the CEC policy, in terms of the structure of the primal optimal solutions $y^*$ of the LP model (2). We assume the BPC policy is well defined, in that the dual optimal solution is unique. We also assume that the dual basis is not the same for $LP(n, D^{c-1})$ and $LP(n - A^j, D^{c-1})$; if the dual basis does not change, then the policies are identical. The following two propositions state and prove these results formally.

**Proposition 4 (Structural Properties of the BPC Policy).** At any state $(n, t)$, if $LP(n, D^{c-1})$ has a unique dual optimal solution, then the corresponding bid-price policy accepts only classes $j$ for which $y^*_j > 0$ in some primal optimal solution.

**Proof.** By analyzing the primal and dual LP, one can distinguish the following situations:

- In all optimal LP-solutions $y^*_j = 0$, then $u^{n, t}_j = 0$ from complementary slackness. By strict complementary slackness (see Bertsimas and Tsitsiklis 1997, p. 192), we have $(v^{n, t})' \cdot A^j + u^{n, t}_j > R_j$, i.e., $BP_j(n, t) = (v^{n, t})' \cdot A^j > R_j$, in which case the bid-price policy accepts class $j$.

- In all optimal LP-solutions $y^*_j = D_j^{c-1}$, in which case $u^{n, t}_j = (R_j - (v^{n, t})' \cdot A^j)^+$ > 0, so the bid-price policy accepts class $j$.

- There is some optimal LP-solution such that $0 < y^*_j < D_j^{c-1}$, which implies that the dual constraint $(v^{n, t})' \cdot A^j - u^{n, t}_j \geq R_j$ is binding and $u^{n, t}_j = 0$, so $(v^{n, t})' \cdot A^j = R_j$, and thus the bid-price policy accepts class $j$. □

**Proposition 5 (Structural Properties of the CEC Policy).** Suppose that $LP(n - A^j, D^{c-1})$ and $LP(n, D^{c-1})$ have different optimal dual bases. Then:
a given request in the same state;

(b) CEC rejects class \( j \) if \( y_j^* < \min(D_j^{-1}, 1) \) in all optimal solutions of \( \text{LP}(n, D^{-1}) \).

**Proof.** If in some optimal solution \( y^* \) of \( \text{LP}(n, D^{-1}) \) we have that \( y_j^* \geq \min(D_j^{-1}, 1) \), then \( y^* - \min(D_j^{-1}, 1) \cdot e_j \geq 0 \) is a feasible solution of \( \text{LP}(n - A', D^{-1}) \), where \( D^{-1} = D^{-1} - \min(D_j^{-1}, 1) \cdot e_j \), and hence,

\[
\text{LP}(n, D^{-1}) = R' \cdot y^* \leq \min(D_j^{-1}, 1) R_j + \text{LP}(n - A', D^{-1}) \leq R_j + \text{LP}(n - A', D^{-1}),
\]

where the last inequality holds because any optimal primal solution of \( \text{LP}(n - A', D^{-1}) \) is feasible to \( \text{LP}(n - A', D^{-1}) \). So, \( \text{OCLP}^D(n, D) = \text{LP}(n, D^{-1}) - \text{LP}(n - A', D^{-1}) \leq R_j \). This proves Part (a).

For Part (b), it follows that \( 0 \leq y_j^* < D_j \) in all primal optimal solutions. By complementary slackness, we have that \( u_{n,j}^{n-1} = 0 \) and \( (v^{n-1})' A' = R_j \). Under the assumption that the optimal dual basis changes, we obtain by Proposition 3 that \( \text{OCLP}^D(n, t) > \text{BP}(n, t) = (v^{n-1})' A' = R_j \), that is CEC rejects class \( j \), which concludes the proof. \( \square \)

### 4.1. An Example

For single-leg instances of the NRM problem, the bid pricing and the CEC algorithms are the same. This is because under the CEC algorithm, the opportunity cost estimates are the same (no change of basis occurs in Equation 4). However, this is not true for the general network case. In this section, we provide an example that highlights the differences between the BPC and the CEC policies, and shows instances where each one is suboptimal. Moreover, we explain why the cross-concavity properties that insure the threshold time structure for the optimal single-leg policy cannot be extended to the network case. We will use the same example to exhibit the following situations:

- An instance when BPC accepts, but CEC rejects a given request in the same state;
- An instance when BPC is suboptimal;
- An instance when CEC is suboptimal;  
- A counterexample of a cross-concavity property (“decreasing differences”) of the LP and DP-value functions for the NRM problem.

In general, if at a given state the CEC policy accepts a request from itinerary \( j \), then at the same state the bid-pricing control policy will also accept, but not vice versa (see Proposition 3). The following situation may occur: \( \text{BP}(j) \leq R_j < \text{OCLP}^D(j) \), and so will accept under the BPC policy but not under the CEC policy.

The following is an example of such behavior, that provides insight into the structural properties of the two policies.

Consider a network with four nodes: a hub \( h \), two origin nodes \( o_1, o_2 \), and a destination node \( d \). The legs of the network are (1) \((o_1, h)\), (2) \((o_2, h)\), (3) \((h, d)\). One can think of this as part of a bigger network where the other (connecting) flights have been sold out. Suppose there is demand from the origin nodes \( o_1, o_2 \) to the hub node \( h \) and to the destination node \( d \). Suppose that there is only one fare class per itinerary, and there is no demand from \( h \) to \( d \). We assume the pricing structure is such that subitineraries cost less: \( R_1 < R_{13}, R_2 < R_{23} \). Furthermore, assume with almost no loss of generality that \( R_{13} + R_2 < R_{23} + R_1 \) (the other case is symmetric, unless equality holds).

Suppose that the available capacity in the current state \( t \) is \( n = (n_1, n_2, n_3) \) and the expected demand for each fare class in the remaining \( t - 1 \) periods is positive. The static LP and its dual can be formulated as follows:

\[
\text{LP}(n, D) = \max_{y, u, v} R_1 y_1 + R_2 y_2 + R_{13} y_{13} + R_{23} y_{23} \\
\text{s.t.} \quad y_1 + y_{13} \leq n_1 \quad y_2 + y_{23} \leq n_2 \quad y_{13} + y_{23} \leq n_3 \quad 0 \leq y \leq D
\]

\[
= \min_{u, v} n v + u D \\
\text{s.t.} \quad v_1 + u_1 \geq R_1 \quad v_2 + u_2 \geq R_2 \quad v_1 + v_3 + u_{13} \geq R_{13} \quad v_2 + v_3 + u_{23} \geq R_{23} \quad u, v \geq 0.
\]

Suppose that there is one seat left on each leg, so \( n = (1, 1, 1) \), and \( D_1 > 1 \) and \( D_{23} > 1 \), so that the demand for the high-paying mix is large enough for the corresponding constraints to be nonbinding in an optimal solution. Then the optimal LP solution is \( y_1^* = y_{23}^* = 1, y_2^* = y_{13}^* = 0 \), and the value of the LP is \( R_1 + R_{23} \).
Since the demand constraints are nonbinding, we must have that \( u = 0 \), and hence \( v_1 \geq R_1, v_2 \geq R_2, v_3 \geq \max(0, R_{13} - v_1, R_{23} - v_2) \). Therefore, in an optimal solution, the shadow prices are equal to \( v_1 = R_1, v_2 = R_2, v_3 = R_{23} - R_2 \). We can compute the bid prices and opportunity costs as follows:

\[
\begin{align*}
OC_{1}^{\text{LP}} &= LP(1, 1, 1, D) - LP(0, 1, 1, D) \\
&= R_1 = BP(1),
\end{align*}
\]

\[
\begin{align*}
OC_{2}^{\text{LP}} &= LP(1, 1, 1, D) - LP(1, 0, 1, D) \\
&= R_1 + R_{23} - R_{13} > BP(2) = R_2,
\end{align*}
\]

\[
\begin{align*}
OC_{13}^{\text{LP}} &= LP(1, 1, 1, D) - LP(0, 1, 0, D) \\
&= R_1 + R_{23} - R_2 = BP(13) > R_{13},
\end{align*}
\]

\[
\begin{align*}
OC_{23}^{\text{LP}} &= LP(1, 1, 1, D) - LP(1, 0, 0, D) \\
&= R_{23} = BP(23).
\end{align*}
\]

Therefore, the two policies disagree on the acceptance of class (2): Under the BPC policy, we will accept, whereas under the CEC policy, we will reject a class (2) request at state \( n = (1, 1, 1) \) as long as there is sufficient forthcoming demand for classes (1) and (23).

The question is which one of the policies is better? Clearly, in the case when demand is deterministic, the BPC policy is suboptimal, by giving away at time \( t \) one unit of capacity that would bring higher revenues in the future. The CEC policy, however, is by definition optimal in the deterministic case because it is equivalent to the DP (certainty equivalence). In the stochastic case, the bid-pricing policy is suboptimal when there is sufficient demand to come from the high-fare mix. Otherwise, we may be better off accepting class (2) right away, in which case our policy is suboptimal.

We can also observe on this example that the LP, and thus the DP value do not exhibit a certain type of cross-concavity property called decreasing differences (see Karaesman and van Ryzin 1998):

**Definition 1.** A function \( f: S \subseteq \mathbb{R}^n \rightarrow \mathbb{R} \) satisfies decreasing differences on \( S \) if for any \( s \in S \), and \( i \neq j \in \{1, \ldots, n\} \), and for all \( \delta_i, \delta_j > 0 \) with \( s + \delta_i e_i + \delta_j e_j \) and \( s + \delta_i e_i - \delta_j e_j \in S \), the following relation holds:

\[
f(s + \delta_i e_i + \delta_j e_j) - f(s + \delta_i e_i) \leq f(s + \delta_i e_i) - f(s).
\]

This cross-concavity property reduces in the univariate case to concavity. This is the key observation underlying the proof of the threshold times property for the optimal single-leg policy (see Diamond and Stone 1991), and would provide a sufficient condition for the property to extend to the network case.

However, the decreasing differences property is violated in our example:

\[
\begin{align*}
LP(1, 1, 1) &= \max(R_1 + R_{23}, R_2 + R_{13}) - R_{23} < R_{13} \\
&= LP(1, 0, 1) - LP(0, 0, 1).
\end{align*}
\]

Notice that the assumption here is that “subitinerary” fares are cheaper \( R_1 < R_{13}, R_2 < R_{23} \). Network effects imply that the opportunity cost of Itinerary (13) under the CEC policy decreases with capacity. Furthermore, if the demand is large enough, the above relation transfers to the corresponding DP values because the two are asymptotically equal. Surprisingly, this says that incremental revenues (opportunity costs) may decrease by decreasing capacity along a certain direction (see also Feng and Lin 2000).

5. Cancellations and Overbooking

The control policies we have discussed so far do not take into account the fact that a significant fraction of customers cancel their reservations during the booking period (cancellations) or simply do not utilize their reservation (no-shows). In these cases customers get full, partial, or no refund, depending on the fare category. In either case, extra capacity becomes available and could be used to accommodate other potential customers. To counterbalance this phenomenon, a common revenue management practice is overbooking. Airlines, hotels, and so on, oversell their inventories to account for reservations that will not materialize. If at the end of the horizon, more demand has materialized than the inventory can accommodate, companies practice class upgrades, pay overbooking penalties to unsatisfied reservations or even perform aircraft changes.

The typical overbooking method practiced by airlines, hotels, and so on, is to decide an initial allocation of overbooking pads, which are virtual increases in leg-capacity. This is usually performed, in practice as well as in the literature, as a static, one-time
incorporate cancellations, no shows and overbooking. In the following, we propose a new method for dynamic overbooking control, where oversales decisions are dynamic and implicit in the admission control mechanism.

5.1. General Models and Relationships
First, we extend the various models described in §2 to incorporate cancellations, no shows and overbooking.

A Dynamic Programming Model. In the case of perfect state information, this model provides the optimal control policy. By allowing cancellations, the state space of the DP-model becomes much larger because it is necessary to keep track of the past sales record s. The random quantities involved are the demand, cancellations, and no-show processes. Given the initial network inventory N, the maximum expected revenue (less refunds and penalties) to be collected from state (s, t) onward (cost-to-go), is given by

\[
DP_N^w(s, t) = \sum_j p_j^c \cdot \max(DP_N^w(s, t-1), \\
R_j + DP_N^w(s + e_j, t-1)) \\
+ \sum_{j\in J} p_j^{nf} \cdot (DP_N^w(s - e_j, t-1) - R_j)
\]

In the following, we propose a new method for dynamic overbooking control, where oversales decisions are dynamic and implicit in the admission control mechanism.

The Integer and Linear Programming Models. In the case of the NRM problem with cancellations and no shows, we observe that both the final revenue gained from, and capacity occupied by a class j reservation are not deterministic quantities because they depend on cancellations and no-shows which are random processes. To define a model that is consistent with these observations, we define \(\tilde{R}_j\) and \(\tilde{A}^j\) to be the expected revenue gained from, and expected capacity occupied by a class j reservation, before the overbooking period. Let \(p_j^c\) and \(p_j^{nf}\) denote the probability that a given class j reservation is cancelled at some point in the booking period, and respectively does not show up for the flight. We assume that these quantities are independent on the time the reservation was made, and so is the cancellation penalty. We denote the revenue expected (or "adjusted") from booking a class j customer as \(\tilde{R}_j = R_j - (1 - p_j^{nf}) \cdot p_j^c \cdot R_j^{nf} - (1 - p_j^c) \cdot p_j^{nf} \cdot R_j^c\), which accounts for potential cancellation and no-shows events and respective refunds \(R_j^{nf}\), \(R_j^c\) (but not overbooking penalties). Let \(\tilde{A}^j = (1 - p_j^c) \cdot (1 - p_j^{nf}) \cdot A^j\) denote the expected capacity occupied by a class j reservation at the end of the horizon. Finally \(\tilde{C}_j = (1 - p_j^c) \cdot (1 - p_j^{nf}) \cdot C_j\) is the average overbooking cost of one class j reservation.

With these notations, we can formulate the following integer programming approximation model, that maximizes expected revenues subject to expected capacity constraints:

\[
IP_N^w(s, t) = \max \ 
\tilde{R} \cdot y - \tilde{A}^j \cdot z^w \\
\text{s.t.} \ \ 0 \leq \tilde{A} \cdot (y + s) - A \cdot z^w \leq N \\
\ \ 0 \leq y \leq D^t \\
\ \ y, z^w \text{ integer.}
\]

The vector y decides how many requests to be accepted in the future, whereas \(z^w\) determines which
customers (itinerary requests) should be bumped at the end of the horizon, if necessary. The “virtual” overbooking pad for each leg is \( A \cdot z^p \). The capacity constraint requires that cancellation-adjusted past and future sales, less oversales, should not exceed the initial network capacity.

With the change of variable \( s^j = z^p_j/(1 - p_j^0) \cdot (1 - p_j^p) \), \( j = 1, \ldots, n \), the corresponding linear programming relaxation, and its dual, can be written as follows:

\[
LP_N^*(s, t) = \max \ R' \cdot y - \tilde{C} \cdot s^p \\
\text{s.t.} \quad \tilde{A} \cdot (s + y - s^p) \leq N \\
\quad \quad 0 \leq y \leq D^t \\
\quad \quad 0 \leq s^p \leq y + s
\]

\[
= \min \ R' - u^* \cdot s^p + (\tilde{R} - u^*)^+ \cdot D^t \\
\text{s.t.} \quad u^* = \min(\tilde{C}, v^* \cdot \tilde{A}) \\
\quad \quad v \geq 0.
\]

5.2. Adjusted Policies

Following the spirit of the basic adaptive bid pricing and CEC policies, we adjust these to incorporate cancellations and overbooking.

Adjusted BPC Policy. We modify the DP model so that at any given state \( S \) a class \( j \) request is accepted if and only if its “adjusted” fare \( \tilde{R}_j \) is higher than either its adjusted bid price, or the adjusted overbooking penalty \( \tilde{C}_j \), i.e.,

\[
\tilde{R}_j \geq \min(\tilde{C}_j, BP_j^p(S)).
\]

Here, bid prices are computed as \( BP_j^p(S) = (v^p) \cdot \tilde{A}^j \), where \( v^p \) represent shadow prices for the leg-capacity constraints in \( LP^*(S) \).

Adjusted CEC Policy. Similarly, we can adapt the CEC policy to accept a class \( j \) request if and only if its “adjusted” revenue exceeds its “adjusted” opportunity cost. \( \tilde{R}_j \geq OC_j^p(S) = LP_N^*(s, t - 1) - LP_N^*(s + e_j, t - 1) \). Again, adjustment accounts for the fact that capacity and revenue might not be realized or may be overbooked.

The same type of arguments can be used to extend Proposition 1 for the case of cancellations, no shows and overbooking. Moreover, the structural properties proved in §4 are preserved in the overbooking-adjusted policies.

6. Computational Results

In this section, we present computational results that illustrate the relative practical performance of the previously described admission control policies for the NRM problem, under different demand regimes. Our objective is to evaluate the different models and policies proposed, in terms of the following criteria: (1) running time, (2) quality of approximation, and (3) robustness. In addition, we would like to further investigate the effectiveness of several extensions and improvements. In §6.1, we report computational results without cancellations and overbooking, while in §6.2, we allow cancellations and overbooking.

6.1. Computational Performance Without Cancellations and Overbooking

We performed expected value calculations for two-leg instances and simulation runs for larger networks. To have a consistent and fairly accurate base for comparing various policies, these were simulated simultaneously on the same realizations of the demand process. The computations were performed in MATLAB 4.0 on an Intel Pentium II Celeron 450 MHz (128 MB RAM, WinNT 4.0). We restricted our attention to smaller instances with the explicit objective to obtain insights into the behavior of both BPC and CEC.

6.1.1. Models and Assumptions. We considered the following types of networks:

N2. A two-leg network with nodes \( o, h, d \), legs (1) \( oh \) (2) \( hd \) and capacities \( N = (N_1, N_2) \). The available itineraries are (1) \( oh \), (2) \( hd \), and (3) \( ohd \), with one class per itinerary. The leg-class incidence matrix, together with the fare structure \( R \), is:

\[
\begin{pmatrix}
R_1 & R_2 & R_3 \\
1 & 0 & 1 \\
0 & 1 & 1
\end{pmatrix}.
\]

N3. The three-leg network described in the example of §4.1, with four nodes: a hub \( h \), two origin nodes \( o_1, o_2 \), and a destination node \( d \), and the legs (1) \( o_1 h \), (2) \( o_2 h \), (3) \( hd \). There is demand from the origin nodes.
\( o_1, o_2 \) to the hub node \( h \) and to the destination node \( d \), on the itineraries: (1) \( o_1 h \), (2) \( o_2 h \), (13) \( o_1 d \), (23) \( o_2 d \). Suppose that there is only one fare class per itinerary, and there is no demand from \( h \) to \( d \). The leg-class incidence matrix, together with the fare structure \( R \), is:

\[
(R) = \begin{pmatrix} R_1 & R_2 & R_{13} & R_{23} \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}.
\]

N4. A four-leg network, with two origins \( o_1, o_2 \), two destinations \( d_1, d_2 \) and a hub \( h \). The legs in the network are (1) \( o_1 h \), (2) \( o_2 h \), (3) \( h d_1 \), (4) \( h d_2 \), with capacities \( N = (N_1, N_2, N_3, N_4) \). There is demand for all the eight itineraries, with one fare class per itinerary. The leg-class incidence matrix, together with the fare structure \( R \), is:

\[
(R) = \begin{pmatrix} R_1 & R_2 & R_3 & R_4 & R_{13} & R_{14} & R_{23} & R_{24} \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}.
\]

N4.2. The same as (N4), except there are two fares per itinerary, so 16 demand classes in all. The leg-class incidence matrix, together with a high-low fare structure \( R = (R^h, R^l) \), is:

\[
(R) = \begin{pmatrix} R^l_1 & R^h_1 & R^l_2 & R^h_2 & R^l_3 & R^h_3 & R^l_4 & R^h_4 & R^l_{13} & R^h_{13} & R^l_{14} & R^h_{14} & R^l_{23} & R^h_{23} & R^l_{24} & R^h_{24} \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \end{pmatrix}.
\]

We used the following alternative scenarios to model the arrival process:

- **HP.** Homogeneous Poisson arrivals with arrival rates given by a constant vector \( \mathbf{p} \) (i.i.d. Bernoulli trials).
- **NHL.** Nonhomogeneous Poisson with high-low demand. We assume arrival rates increase for high-fare classes \((p^l = p / \log(a + (T-t) \cdot \rho)) \) and decrease \((p^l = p / \log(a + (T-t) \cdot \rho)) \) for low-fare classes, as we approach departure.
- **NLH.** Nonhomogeneous Poisson with low-high demand. We assume arrival rates decrease for high-fare classes \((p^l = p / \log(a + (T-t) \cdot \rho)) \) and increase \((p^l = p / \log(a + (T-t) \cdot \rho)) \) for low-fare classes, as we approach departure.

The log-factors are given by \( \rho \), and \( a \) is a constant, equal to the base of the logarithm (we take \( a = 2 \)).

6.1.2. **Running Time.** Exact calculations of the optimal expected revenue (DP), and expected values of the proposed policies (CEC, BPC) are practically impossible. Already for two-leg networks (N2) with one class per itinerary, 50 seats per leg initial capacity and \( T = 300 \) time periods the computation takes about two hours.

A tractable approach for measuring performance of the proposed policies, however, is provided by simulation. A simulation run of the CEC or BPC policies for a two-leg network takes less than a minute (less than a second per iteration). The largest network we simulated was a four-leg network (N4.2) with two demand classes per itinerary. When the initial capacities are in the order of \( N = 50–100 \) and the time horizon has \( T = 300 \) time periods, a full simulation run for such an instance takes a few minutes (i.e., a few seconds per iteration).
Table 2 Expected Value Calculation for (N2) with N = (50, 50); R = (25, 20, 35); p = (0.4, 0.3, 0.1); T = 10–200

<table>
<thead>
<tr>
<th>T</th>
<th>LP</th>
<th>DP</th>
<th>CEC</th>
<th>BPC</th>
<th>LP</th>
<th>DP</th>
<th>CEC</th>
<th>BPC</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>195</td>
<td>195</td>
<td>195</td>
<td>195</td>
<td>201.4</td>
<td>201.4</td>
<td>201.4</td>
<td>201.4</td>
</tr>
<tr>
<td>20</td>
<td>390</td>
<td>390</td>
<td>390</td>
<td>390</td>
<td>413.5</td>
<td>413.5</td>
<td>413.5</td>
<td>413.5</td>
</tr>
<tr>
<td>30</td>
<td>585</td>
<td>585</td>
<td>585</td>
<td>585</td>
<td>632.9</td>
<td>632.9</td>
<td>632.9</td>
<td>632.9</td>
</tr>
<tr>
<td>40</td>
<td>780</td>
<td>780</td>
<td>780</td>
<td>780</td>
<td>857.6</td>
<td>857.6</td>
<td>857.6</td>
<td>857.6</td>
</tr>
<tr>
<td>50</td>
<td>975</td>
<td>975</td>
<td>975</td>
<td>975</td>
<td>1,086.5</td>
<td>1,086.5</td>
<td>1,086.5</td>
<td>1,086.5</td>
</tr>
<tr>
<td>60</td>
<td>1,170</td>
<td>1,170</td>
<td>1,170</td>
<td>1,170</td>
<td>1,318.6</td>
<td>1,318.6</td>
<td>1,318.6</td>
<td>1,318.6</td>
</tr>
<tr>
<td>70</td>
<td>1,365</td>
<td>1,365</td>
<td>1,365</td>
<td>1,365</td>
<td>1,553.4</td>
<td>1,552.2</td>
<td>1,552.1</td>
<td>1,552.2</td>
</tr>
<tr>
<td>80</td>
<td>1,560</td>
<td>1,559.5</td>
<td>1,559.5</td>
<td>1,559.5</td>
<td>1,790.4</td>
<td>1,755.3</td>
<td>1,754.5</td>
<td>1,754.9</td>
</tr>
<tr>
<td>90</td>
<td>1,755</td>
<td>1,745.7</td>
<td>1,745.4</td>
<td>1,745.7</td>
<td>1,885.3</td>
<td>1,863.3</td>
<td>1,860.8</td>
<td>1,840.9</td>
</tr>
<tr>
<td>100</td>
<td>1,950</td>
<td>1,897.5</td>
<td>1,896.4</td>
<td>1,897.1</td>
<td>1,925.1</td>
<td>1,904.7</td>
<td>1,901.4</td>
<td>1,852.1</td>
</tr>
<tr>
<td>110</td>
<td>2,020</td>
<td>1,998.9</td>
<td>1,986.6</td>
<td>1,993.4</td>
<td>1,937.3</td>
<td>1,922.1</td>
<td>1,917.1</td>
<td>1,863.3</td>
</tr>
<tr>
<td>120</td>
<td>2,090</td>
<td>2,064.6</td>
<td>2,061.6</td>
<td>2,031.6</td>
<td>1,949.1</td>
<td>1,934.3</td>
<td>1,927.1</td>
<td>1,876.9</td>
</tr>
<tr>
<td>130</td>
<td>2,140</td>
<td>2,110.6</td>
<td>2,107.3</td>
<td>2,021.9</td>
<td>1,969.0</td>
<td>1,945.9</td>
<td>1,936.7</td>
<td>1,889.5</td>
</tr>
<tr>
<td>140</td>
<td>2,170</td>
<td>2,145.5</td>
<td>2,140.5</td>
<td>2,018.3</td>
<td>1,971.7</td>
<td>1,957.1</td>
<td>1,946.2</td>
<td>1,899.5</td>
</tr>
<tr>
<td>150</td>
<td>2,200</td>
<td>2,175.7</td>
<td>2,167.6</td>
<td>2,018.9</td>
<td>1,982.5</td>
<td>1,968.1</td>
<td>1,955.6</td>
<td>1,907.2</td>
</tr>
<tr>
<td>160</td>
<td>2,230</td>
<td>2,202.4</td>
<td>2,192.9</td>
<td>2,019.3</td>
<td>1,993.1</td>
<td>1,978.7</td>
<td>1,964.9</td>
<td>1,913.1</td>
</tr>
<tr>
<td>170</td>
<td>2,250</td>
<td>2,222.8</td>
<td>2,216.9</td>
<td>2,019.4</td>
<td>2,003.4</td>
<td>1,989.1</td>
<td>1,974.0</td>
<td>1,918.1</td>
</tr>
<tr>
<td>180</td>
<td>2,250</td>
<td>2,236.2</td>
<td>2,233</td>
<td>2,019.4</td>
<td>2,013.5</td>
<td>1,999.3</td>
<td>1,983.0</td>
<td>1,922.9</td>
</tr>
<tr>
<td>190</td>
<td>2,250</td>
<td>2,243.8</td>
<td>2,242.3</td>
<td>2,019.4</td>
<td>2,023.5</td>
<td>2,009.3</td>
<td>1,991.8</td>
<td>1,926.6</td>
</tr>
<tr>
<td>200</td>
<td>2,250</td>
<td>2,247.5</td>
<td>2,246.9</td>
<td>2,019.4</td>
<td>2,033.2</td>
<td>2,019</td>
<td>2,000.5</td>
<td>1,934.7</td>
</tr>
</tbody>
</table>

**Note.** The first section considers homogeneous arrivals, and the next two are nonhomogeneous arrivals with p = (50, 50, 30) and a = 2.

For more realistic airline networks [20 legs (10 origins, 10 destinations, and one hub), 600 classes (100 itineraries, 6 fare classes)] the time per iteration is still in the order of a few minutes. This supports the idea that these dynamic policies can be used in an online fashion to provide an adaptive control mechanism where opportunity cost estimates are rapidly recalculated at each iteration.

6.1.3. Quality of Approximation

**Expected Value Computations for Two-Leg Networks.** The results in Table 2 show expected value calculations for the case of the two-leg network, with different initial capacities, fare structures, and demand regimes. In all cases, we observe that when the time horizon is large, the value of the CEC policy is near optimal, whereas the bid-pricing policy appears to be off by a constant. When the time horizon is small, both policies perform optimally. For intermediate stages when both some demand and capacity constraints become binding, it is not clear which policy is better. Note that the BPC value is non-monotonic over time. We believe that this is because of the dependence of the policy on the choice of dual solutions (not enforced in our program).

**Simulation Results.** For two-leg networks we provide in Table 3 a comparison between simulation results and exact calculations for the DP value function and expected values of the CEC and BPC policies. The high dimensionality of the problem does not allow for exact expected value calculations for larger networks. In Table 4, we present results from...
noise in demand forecasts, we compare LP(n, D) and $E_\delta[LP(n, D + \delta)]$. From the dual formulation we have that

$$E_\delta[LP(n, D + \delta)] - LP(n, D) = E_\delta[OC_\delta(n, D)] \leq (R' - v' \cdot A)^+ \cdot E[\delta],$$

where $v$ are the leg-shadow prices of LP(n, D). So there is a sublinear effect on the value function, associated with the forecasting bias of the demand for those classes whose fares exceed their bid prices.

Furthermore, computational experiments show that the CEC policy is surprisingly robust to certain forms of noise and bias in the demand data, much more so than BPC.

### Correlated Random Noise.

At each point in time $t$ we generate a multivariate random variable $\delta \sim N(b, \sigma)$, to introduce noise into the arrival forecast. We propose two alternatives for adding noise to the forecast:

- **Constant rate noise**: Generate $\delta$ and add it to the arrival rate at each time;
- **Log-rate noise**: Generate $\delta$ and add $\delta' = \log(a + p \cdot (t - a)/T) \cdot \hat{\delta}$ to the arrival rate.

From the results in Table 5, we observe that CEC is constantly more robust than BPC to noise and bias. The columns LP$_\delta$ compute the LP-values with noisy forecasts: LP$_\delta(n, D') = LP(n, D' + E[\delta'])$.

### Uncorrelated Random Noise.

The goal of this computational exercise is to illustrate and compare the robustness of the BPC and CEC policies, to increasing bias in the demand forecast. We start with an initial (uncorrelated) bias per arrival $b$, and in each consecutive experiment subscripted $(ab)$, we increase the bias by a factor of $\alpha = 1, 2, 4$. In the last experiment ($\delta$), we introduce correlated noise ($\delta$ same as above) in the arrival rate estimation. We observe from Table 6 that CEC is constantly more robust than BPC to noise and bias.

### 6.1.5. Extensions

**Rollout Heuristics.** We examine how previously computed values of the CEC and BPC policies can be used to provide estimates of opportunity costs, leading to the roll-out heuristics R(CEC) and R(BPC). We

<table>
<thead>
<tr>
<th>Table 4</th>
<th>Simulations for Larger Networks (Top-Down)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T$</td>
<td>LP ($σ$)</td>
</tr>
<tr>
<td>(N3)</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>2,300</td>
</tr>
<tr>
<td>200</td>
<td>3,200</td>
</tr>
<tr>
<td>300</td>
<td>3,600</td>
</tr>
<tr>
<td>(N4.1)</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>3,500</td>
</tr>
<tr>
<td>300</td>
<td>5,800</td>
</tr>
<tr>
<td>600</td>
<td>6,050</td>
</tr>
<tr>
<td>(N4.2)</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>4,070</td>
</tr>
<tr>
<td>200</td>
<td>6,125</td>
</tr>
<tr>
<td>300</td>
<td>11,110</td>
</tr>
</tbody>
</table>

Note. (N3): $N = (60, 60, 100); R = (10, 20, 50, 30)$ and $p = (0.2, 0.3, 0.4)$; (N4.1): $N_j = 50, R = (20, 20, 50, 30, 60, 40, 45, 55)$, $p = (0.2, 0.25, 0.05, 0.05, 0.1, 0.15, 0.1, 0.1);$ (N4.2): $N_j = 50, R = (20, 20, 50, 30, 60, 40, 45, 55)$, with $R = 2 \cdot R$ and $p = (0.15, 0.02, 0.2, 0.01, 0.05, 0.01, 0.1, 0.01, 0.15, 0.015, 0.1, 0.01, 0.1, 0.01).
observe that the rollout procedure produces a significant improvement in the quality of the value function approximation, especially for the BPC policy (10%). This can be observed in Table 7 where we performed calculations for various two-leg networks.

In practice, however, it is too expensive to compute and store all the $H$-values. In order to implement this idea effectively, we suggest storing $H$-values from several insightful a priori simulations, and interpolating these in an online fashion (as needed), for the “second time around” policy. It is an interesting idea to investigate what types of preprocessing simulations would provide an insightful information database.

Simulations Using Monte Carlo Demand Estimation. We run a simplified version of the Monte Carlo policy described in §3.3, that assigns the same weights to all the trials and defines:

$$OC_j^{MC}(n, t) = \frac{1}{r} \sum_{i=1}^{r} OC_j(n, \hat{D}_i^{-1}).$$

When the number of trials $r$ is large enough (so we have a reasonably good MC-demand estimate), this policy provides a visible improvement (1.6%) over the original version, as can be observed from the following computational examples. However, when the number of trials is not large enough (30 or less), we have observed that this policy delivers a poor performance. The tables below provide estimates for the value of different models (LP, PI$^{\text{MP}}$, MC, CEC, BPC), as well as standard deviations, lower and upper bounds obtained through simulation. We also calculate the estimated average ratio $P$ between different models, as well as the standard deviation, minimum, and maximum ratio observed during simulation. One can observe for instance that the ratio CEC/BPC is always $\geq 1$, whereas the ratio MC/PI$^{\text{MP}}$ is surprisingly high, ranging between 96%–99%. The computational results presented in Table 8 show an improvement of up to 20% of MC over CEC.

### Table 5

<table>
<thead>
<tr>
<th>$T$</th>
<th>LP</th>
<th>LP$_a$</th>
<th>DP</th>
<th>CEC</th>
<th>BPC</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>195</td>
<td>213.7</td>
<td>195</td>
<td>195</td>
<td>195</td>
</tr>
<tr>
<td>20</td>
<td>390</td>
<td>416.7</td>
<td>390</td>
<td>390</td>
<td>390</td>
</tr>
<tr>
<td>30</td>
<td>585</td>
<td>611.9</td>
<td>585</td>
<td>585</td>
<td>585</td>
</tr>
<tr>
<td>40</td>
<td>780</td>
<td>784.6</td>
<td>780</td>
<td>780</td>
<td>780</td>
</tr>
<tr>
<td>50</td>
<td>975</td>
<td>975.1</td>
<td>975</td>
<td>975</td>
<td>975</td>
</tr>
<tr>
<td>60</td>
<td>1,170</td>
<td>1,175.5</td>
<td>1,170</td>
<td>1,170</td>
<td>1,170</td>
</tr>
<tr>
<td>70</td>
<td>1,365</td>
<td>1,366.4</td>
<td>1,365</td>
<td>1,365</td>
<td>1,365</td>
</tr>
<tr>
<td>80</td>
<td>1,560</td>
<td>1,605.6</td>
<td>1,559.5</td>
<td>1,559.5</td>
<td>1,559.5</td>
</tr>
<tr>
<td>90</td>
<td>1,755</td>
<td>1,770.3</td>
<td>1,745.7</td>
<td>1,745.2</td>
<td>1,745.3</td>
</tr>
<tr>
<td>100</td>
<td>1,950</td>
<td>1,977.8</td>
<td>1,897.5</td>
<td>1,895.8</td>
<td>1,895.4</td>
</tr>
<tr>
<td>110</td>
<td>2,020</td>
<td>2,086.1</td>
<td>1,998.9</td>
<td>1,996.1</td>
<td>1,961.4</td>
</tr>
<tr>
<td>120</td>
<td>2,090</td>
<td>2,115.2</td>
<td>2,064.6</td>
<td>2,061.7</td>
<td>1,971.7</td>
</tr>
<tr>
<td>130</td>
<td>2,140</td>
<td>2,139.4</td>
<td>2,110.6</td>
<td>2,107.1</td>
<td>1,980.1</td>
</tr>
<tr>
<td>140</td>
<td>2,170</td>
<td>2,172.7</td>
<td>2,145.5</td>
<td>2,139.6</td>
<td>1,987.7</td>
</tr>
<tr>
<td>150</td>
<td>2,200</td>
<td>2,197.8</td>
<td>2,175.7</td>
<td>2,166.9</td>
<td>1,994.5</td>
</tr>
<tr>
<td>160</td>
<td>2,230</td>
<td>2,227.8</td>
<td>2,202.4</td>
<td>2,192.8</td>
<td>1,999.5</td>
</tr>
<tr>
<td>170</td>
<td>2,250</td>
<td>2,250.0</td>
<td>2,228.8</td>
<td>2,216.8</td>
<td>2,000.2</td>
</tr>
<tr>
<td>180</td>
<td>2,250</td>
<td>2,250.0</td>
<td>2,236.2</td>
<td>2,233</td>
<td>2,000.2</td>
</tr>
<tr>
<td>190</td>
<td>2,250</td>
<td>2,250.0</td>
<td>2,243.8</td>
<td>2,242.3</td>
<td>2,130.6</td>
</tr>
<tr>
<td>200</td>
<td>2,250</td>
<td>2,250.0</td>
<td>2,247.5</td>
<td>2,246.9</td>
<td>2,227</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$T$</th>
<th>LP</th>
<th>LP$_a$</th>
<th>DP</th>
<th>CEC</th>
<th>BPC</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>195</td>
<td>221.1</td>
<td>195</td>
<td>195</td>
<td>195</td>
</tr>
<tr>
<td>20</td>
<td>390</td>
<td>408.7</td>
<td>390</td>
<td>390</td>
<td>390</td>
</tr>
<tr>
<td>30</td>
<td>585</td>
<td>623.3</td>
<td>585</td>
<td>585</td>
<td>585</td>
</tr>
<tr>
<td>40</td>
<td>780</td>
<td>765.5</td>
<td>780</td>
<td>780</td>
<td>780</td>
</tr>
<tr>
<td>50</td>
<td>975</td>
<td>957.2</td>
<td>975</td>
<td>975</td>
<td>975</td>
</tr>
<tr>
<td>60</td>
<td>1,170</td>
<td>1,123.5</td>
<td>1,170</td>
<td>1,170</td>
<td>1,170</td>
</tr>
<tr>
<td>70</td>
<td>1,365</td>
<td>1,315.3</td>
<td>1,365</td>
<td>1,365</td>
<td>1,365</td>
</tr>
<tr>
<td>80</td>
<td>1,560</td>
<td>1,569</td>
<td>1,559.5</td>
<td>1,559.5</td>
<td>1,559.5</td>
</tr>
<tr>
<td>90</td>
<td>1,755</td>
<td>1,685.5</td>
<td>1,745.7</td>
<td>1,745.2</td>
<td>1,745.1</td>
</tr>
<tr>
<td>100</td>
<td>1,950</td>
<td>1,878.5</td>
<td>1,897.5</td>
<td>1,895.8</td>
<td>1,894.6</td>
</tr>
<tr>
<td>110</td>
<td>2,020</td>
<td>2,088.5</td>
<td>1,998.9</td>
<td>1,996</td>
<td>1,974.7</td>
</tr>
<tr>
<td>120</td>
<td>2,090</td>
<td>2,069.2</td>
<td>2,064.6</td>
<td>2,061.6</td>
<td>1,967.2</td>
</tr>
<tr>
<td>130</td>
<td>2,140</td>
<td>2,105.5</td>
<td>2,110.6</td>
<td>2,106.3</td>
<td>1,981.2</td>
</tr>
<tr>
<td>140</td>
<td>2,170</td>
<td>2,147.1</td>
<td>2,145.5</td>
<td>2,137.4</td>
<td>1,987.9</td>
</tr>
<tr>
<td>150</td>
<td>2,200</td>
<td>2,168</td>
<td>2,175.7</td>
<td>2,162.5</td>
<td>1,993.5</td>
</tr>
<tr>
<td>160</td>
<td>2,230</td>
<td>2,178.8</td>
<td>2,202.4</td>
<td>2,185.5</td>
<td>2,030.2</td>
</tr>
<tr>
<td>170</td>
<td>2,250</td>
<td>2,203.2</td>
<td>2,222.8</td>
<td>2,208.4</td>
<td>2,151.1</td>
</tr>
<tr>
<td>180</td>
<td>2,250</td>
<td>2,240.5</td>
<td>2,236.2</td>
<td>2,227.3</td>
<td>2,208.9</td>
</tr>
<tr>
<td>190</td>
<td>2,250</td>
<td>2,235.2</td>
<td>2,243.8</td>
<td>2,240.3</td>
<td>2,223.4</td>
</tr>
<tr>
<td>200</td>
<td>2,250</td>
<td>2,248.5</td>
<td>2,247.5</td>
<td>2,246.8</td>
<td>2,230.6</td>
</tr>
</tbody>
</table>

*Note. (N2): $N = (50, 50); R = (25, 20, 35); p = (0.4, 0.3, 0.1); b = (0.07, –0.05, 0.03); $\alpha = \frac{1}{5} \begin{pmatrix} 0.1 & -0.02 & 0.04 \\ -0.02 & 0.08 & 0.01 \\ 0.04 & 0.01 & 0.06 \end{pmatrix}.$*
Table 6 Increasing Bias in Arrivals

<table>
<thead>
<tr>
<th>T</th>
<th>DP</th>
<th>LP</th>
<th>CEC</th>
<th>BPC</th>
<th>LP₁₉</th>
<th>CEC₁₉</th>
<th>BPC₁₉</th>
<th>LP₂₈</th>
<th>CEC₂₈</th>
<th>BPC₂₈</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>190</td>
<td>190</td>
<td>190</td>
<td>190</td>
<td>196.227</td>
<td>190</td>
<td>190</td>
<td>202.454</td>
<td>190</td>
<td>190</td>
</tr>
<tr>
<td>20</td>
<td>380</td>
<td>380</td>
<td>380</td>
<td>380</td>
<td>387.902</td>
<td>380</td>
<td>380</td>
<td>395.812</td>
<td>380</td>
<td>380</td>
</tr>
<tr>
<td>30</td>
<td>568.0307</td>
<td>570</td>
<td>567.967</td>
<td>568.0053</td>
<td>578.9176</td>
<td>567.958</td>
<td>568.0149</td>
<td>587.8351</td>
<td>567.9583</td>
<td>568.0271</td>
</tr>
<tr>
<td>40</td>
<td>713.9429</td>
<td>740</td>
<td>712.8882</td>
<td>711.4275</td>
<td>751.7866</td>
<td>712.6471</td>
<td>711.3734</td>
<td>763.5732</td>
<td>712.654</td>
<td>710.6125</td>
</tr>
<tr>
<td>50</td>
<td>784.8309</td>
<td>815</td>
<td>782.7964</td>
<td>745.0979</td>
<td>818.9707</td>
<td>782.1006</td>
<td>743.3085</td>
<td>822.9414</td>
<td>781.6062</td>
<td>737.3545</td>
</tr>
<tr>
<td>60</td>
<td>819.8745</td>
<td>845</td>
<td>817.9585</td>
<td>749.6329</td>
<td>849.1515</td>
<td>817.5189</td>
<td>748.0799</td>
<td>853.3030</td>
<td>816.9229</td>
<td>742.664</td>
</tr>
<tr>
<td>70</td>
<td>841.6218</td>
<td>855</td>
<td>839.8851</td>
<td>749.8441</td>
<td>855</td>
<td>839.6463</td>
<td>748.1471</td>
<td>855</td>
<td>839.3091</td>
<td>742.664</td>
</tr>
<tr>
<td>80</td>
<td>851.0686</td>
<td>855</td>
<td>850.2119</td>
<td>749.8441</td>
<td>855</td>
<td>850.1653</td>
<td>748.1471</td>
<td>855</td>
<td>850.1115</td>
<td>742.664</td>
</tr>
<tr>
<td>90</td>
<td>854.0811</td>
<td>855</td>
<td>853.8794</td>
<td>749.8441</td>
<td>855</td>
<td>853.8777</td>
<td>748.1471</td>
<td>855</td>
<td>853.8765</td>
<td>742.664</td>
</tr>
<tr>
<td>99</td>
<td>854.7912</td>
<td>855</td>
<td>854.7523</td>
<td>749.8441</td>
<td>855</td>
<td>854.7523</td>
<td>748.1471</td>
<td>855</td>
<td>854.7523</td>
<td>742.664</td>
</tr>
</tbody>
</table>

Note: (N²): Data: N = (19, 19); R = (25, 20, 35); p = (0.3, 0.4, 0.1); b = (0.07, −0.05, 0.03).

6.2. Computational Performance Under Cancellations and Overbooking

The objective in this section is to understand the relative performance of BPC and CEC in an environment with cancellations and overbooking. We considered a booking horizon of 15 periods for a hub and spoke network with five cities and two classes, of the type (N4.2). The arrival process for the highest fare class is nonhomogeneous Poisson with rate 0.5 for Periods 1–13 and 5 for Periods 14, 15. The arrival process for the lowest fare class is homogeneous Poisson with Rate 3. For simplicity, we kept the fare of the higher class in a single-leg itinerary equal to $100 and of the lower class equal to $80. We varied the fare of two-leg itineraries. After experimentation we identified the following parameters that affected the relative performance of CEC and BPC: We varied the following parameters: (a) The overbooking penalty (Cₙ), (b) the probability of cancellation (pₓ), and (c) the fare of a two-leg itinerary compared to a single-leg itinerary.

The implementation was done in C and the algorithms were run on a Dell Pentium III 600MHz operating under Linux.

In Table 9, we report the behavior of CEC and BPC as a function of the overbooking penalty. We observe that CEC leads to consistently higher revenue by approximately 1%.

In Table 10, we report the behavior of CEC and BPC as a function of the cancellation probability. With the exception of very high cancellation rate (0.30), CEC outperforms BPC.

In Table 11, we report the behavior of CEC and BPC as a function of the ratio p of the fare of a two-leg itinerary versus the fare of a single-leg itinerary. CEC outperforms BPC, but the level of overperformance decreases as p ranges from 1 to 2.

7. Conclusions

In this paper, we have presented several models and algorithms for solving the stochastic and dynamic
but as opposed to BPC, it is more “robust” in the following sense:

- The CEC opportunity cost estimates are uniquely defined at each state, whereas for additive bid prices, there may be several dual optimal solutions.
- The CEC policy is optimal in the deterministic regime, whereas BPC is not.
- Computationally we observe that the CEC policy outperforms the BPC policy when the load factors (the ratio of expected demand to available capacity) tend to be large. When the load factor is small, both policies perform optimally. There is a critical range

NRM problem. We proposed a new efficient algorithm, based on a certainty equivalent approximation and compared it with the widely used bid-price control policy. This policy conceptually improves the current NRM-approach based on additive bid pricing, by using more insightful, piecewise linear approximations of opportunity cost. It is just as easy to compute, but as opposed to BPC, it is more “robust” in the following sense:

\begin{table}
\centering
\begin{tabular}{cccccccc}
\hline
T & LP & DP & CEC & BPC & R(CEC) & R(BPC) \\
\hline
1 & 19.5 & 19.5 & 19.5 & 19.5 & 19.5 & 19.5 \\
10 & 195 & 195 & 195 & 195 & 195 & 195 \\
20 & 390 & 390 & 390 & 390 & 390 & 390 \\
30 & 585 & 585 & 585 & 585 & 585 & 585 \\
40 & 780 & 780 & 780 & 780 & 780 & 780 \\
50 & 975 & 975 & 975 & 975 & 975 & 975 \\
60 & 1,170 & 1,170 & 1,170 & 1,170 & 1,170 & 1,170 \\
70 & 1,365 & 1,365 & 1,365 & 1,365 & 1,365 & 1,365 \\
80 & 1,560 & 1,559.5 & 1,559.5 & 1,559.5 & 1,559.5 & 1,559.5 \\
90 & 1,755 & 1,745.7 & 1,745.4 & 1,745.7 & 1,745.6 & 1,745.7 \\
100 & 1,950 & 1,897.5 & 1,896.4 & 1,897.1 & 1,897.2 & 1,897.4 \\
110 & 2,020 & 1,998.9 & 1,996.6 & 1,993.4 & 1,998 & 1,995.5 \\
120 & 2,090 & 2,046.6 & 2,061.6 & 2,031.6 & 2,063.2 & 2,042.2 \\
130 & 2,140 & 2,110.6 & 2,127.3 & 2,021.9 & 2,108.9 & 2,052.8 \\
140 & 2,170 & 2,145.5 & 2,140.5 & 2,018.3 & 2,143 & 2,104.2 \\
150 & 2,200 & 2,175.7 & 2,167.6 & 2,018.9 & 2,172.9 & 2,163.9 \\
160 & 2,230 & 2,202.4 & 2,192.9 & 2,019.3 & 2,199.6 & 2,197.2 \\
170 & 2,250 & 2,222.8 & 2,216.9 & 2,019.4 & 2,220.6 & 2,217.4 \\
180 & 2,250 & 2,236.2 & 2,233 & 2,019.4 & 2,234.8 & 2,230.2 \\
190 & 2,250 & 2,243.8 & 2,242.3 & 2,019.4 & 2,243.1 & 2,238.1 \\
200 & 2,250 & 2,247.5 & 2,246.9 & 2,019.4 & 2,247.2 & 2,241.9 \\
\hline
\end{tabular}
\caption{Rollout Heuristics}
\end{table}

\textit{Note.} (N2.1): N = (50, 50), R = (25, 20, 35), p = (0.4, 0.3, 0.1); (N2.2): N = (19, 19), R = (25, 20, 35), p = (0.3, 0.4, 0.1).
when the load factor is close to one, where it is not clear which policy provides a better performance, but the difference between the two is very small. Moreover, computational exercises show that the value of the CEC policy is very close to the value function DP, and to the perfect information upper bound PI^{LP}.

- The CEC policy is significantly more robust than BPC to bias and (correlated) noise in the demand forecast.

The following extensions provide insights towards further developments:

- The rollout procedure produces an important improvement in the value of both policies, virtually closing the optimality gap.
- The Monte Carlo simulation procedure incorporates demand variability, and produces a significant improvement in the value of the CEC policy when the number of trials is sufficiently large.

### Table 8

<table>
<thead>
<tr>
<th>(N2.1)</th>
<th>LP</th>
<th>PI^{LP}</th>
<th>MC</th>
<th>CEC</th>
<th>BPC</th>
<th>P</th>
<th>AvgP</th>
<th>StdP</th>
<th>MinP</th>
<th>MaxP</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXP</td>
<td>2,200</td>
<td>2,191.2</td>
<td>2,170</td>
<td>2,162.8</td>
<td>2,161.6</td>
<td>CEC/BPC</td>
<td>1.0006</td>
<td>0.0035</td>
<td>0.9931</td>
<td>1.0074</td>
</tr>
<tr>
<td>Std</td>
<td>0</td>
<td>44.48</td>
<td>50.45</td>
<td>52.41</td>
<td>54.19</td>
<td>CEC/PI^{LP}</td>
<td>0.987</td>
<td>0.0093</td>
<td>0.9569</td>
<td>1</td>
</tr>
<tr>
<td>LB</td>
<td>2,200</td>
<td>2,070</td>
<td>2,035</td>
<td>2,000</td>
<td>2,000</td>
<td>MC/PI^{LP}</td>
<td>0.9903</td>
<td>0.0084</td>
<td>0.9645</td>
<td>1</td>
</tr>
<tr>
<td>UB</td>
<td>2,200</td>
<td>2,250</td>
<td>2,250</td>
<td>2,250</td>
<td>2,240</td>
<td>MC/CEC</td>
<td>1.003</td>
<td>0.0067</td>
<td>0.9909</td>
<td>1.0225</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(N2.2)</th>
<th>LP</th>
<th>PI^{LP}</th>
<th>MC</th>
<th>CEC</th>
<th>BPC</th>
<th>P</th>
<th>AvgP</th>
<th>StdP</th>
<th>MinP</th>
<th>MaxP</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXP</td>
<td>225</td>
<td>215</td>
<td>208</td>
<td>205</td>
<td>202.5</td>
<td>CEC/BPC</td>
<td>1.01</td>
<td>0.028</td>
<td>1</td>
<td>1.075</td>
</tr>
<tr>
<td>Std</td>
<td>0</td>
<td>10.55</td>
<td>16.53</td>
<td>18.7</td>
<td>19.76</td>
<td>CEC/PI^{LP}</td>
<td>0.9521</td>
<td>0.0498</td>
<td>0.8537</td>
<td>1</td>
</tr>
<tr>
<td>LB</td>
<td>225</td>
<td>195</td>
<td>185</td>
<td>175</td>
<td>175</td>
<td>MC/PI^{LP}</td>
<td>0.9664</td>
<td>0.038</td>
<td>0.9024</td>
<td>1</td>
</tr>
<tr>
<td>UB</td>
<td>225</td>
<td>225</td>
<td>225</td>
<td>225</td>
<td>225</td>
<td>MC/CEC</td>
<td>1.016</td>
<td>0.034</td>
<td>1</td>
<td>1.0857</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(N2.3)</th>
<th>LP</th>
<th>PI^{LP}</th>
<th>MC</th>
<th>CEC</th>
<th>BPC</th>
<th>P</th>
<th>AvgP</th>
<th>StdP</th>
<th>MinP</th>
<th>MaxP</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXP</td>
<td>8,000</td>
<td>7,922</td>
<td>7,785</td>
<td>7,771</td>
<td>7,766</td>
<td>CEC/BPC</td>
<td>1.0007</td>
<td>0.011</td>
<td>0.975</td>
<td>1.02</td>
</tr>
<tr>
<td>Std</td>
<td>0</td>
<td>224.3</td>
<td>238.7</td>
<td>285.2</td>
<td>283.8</td>
<td>CEC/PI^{LP}</td>
<td>0.9808</td>
<td>0.0166</td>
<td>0.94</td>
<td>1</td>
</tr>
<tr>
<td>LB</td>
<td>8,000</td>
<td>7,500</td>
<td>7,200</td>
<td>7,050</td>
<td>7,050</td>
<td>MC/PI^{LP}</td>
<td>0.9828</td>
<td>0.017</td>
<td>0.939</td>
<td>1</td>
</tr>
<tr>
<td>UB</td>
<td>8,000</td>
<td>8,600</td>
<td>8,400</td>
<td>8,500</td>
<td>8,400</td>
<td>MC/CEC</td>
<td>1.002</td>
<td>0.0184</td>
<td>0.9615</td>
<td>1.0425</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(N4)</th>
<th>LP</th>
<th>PI^{LP}</th>
<th>MC</th>
<th>CEC</th>
<th>BPC</th>
<th>P</th>
<th>AvgP</th>
<th>StdP</th>
<th>MinP</th>
<th>MaxP</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXP</td>
<td>6,800</td>
<td>n.a.</td>
<td>6,694.4</td>
<td>6,588</td>
<td>6,568.8</td>
<td>CEC/BPC</td>
<td>1.0029</td>
<td>0.0051</td>
<td>0.9935</td>
<td>1.0117</td>
</tr>
<tr>
<td>Std</td>
<td>0</td>
<td>n.a.</td>
<td>214.44</td>
<td>213.73</td>
<td>215</td>
<td>MC/CEC</td>
<td>1.01624</td>
<td>0.0127</td>
<td>0.9908</td>
<td>1.0421</td>
</tr>
<tr>
<td>LB</td>
<td>6,800</td>
<td>n.a.</td>
<td>6,260</td>
<td>6,140</td>
<td>6,160</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UB</td>
<td>6,800</td>
<td>n.a.</td>
<td>7,065</td>
<td>6,950</td>
<td>6,930</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: \( N = (50, 50) \), \( R = (25, 20, 35) \), \( p = (0.3, 0.4, 0.1) \), \( T = 150, NT = 100, r = 50 \); \( N = (5, 5) \), \( R = (25, 20, 35) \), \( p = (0.3, 0.4, 0.1) \), \( T = 20, NT = 10, r = 50 \); \( N = (20, 20) \), \( R = (25, 20, 35) \), \( p = (0.2, 0.3, 0.5) \), \( T = 50, NT = 50, r = 50 \); \( N = (50, 50, 100, 100) \), \( R = (20, 20, 30, 30, 60, 40, 45, 55) \), \( p = (0.15, 0.15, 0.15, 0.15, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1) \), \( T = 200, NT = 50, r = 60 \).
Table 11 The Expected Revenue in 200 Simulation Runs as a Function of the Ratio \( \rho \) of the Fare of a Two-Leg Itinerary Versus the Fare of a Single-Leg Itinerary

<table>
<thead>
<tr>
<th>( \rho )</th>
<th>CEC</th>
<th>BPC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>22,522</td>
<td>20,587</td>
</tr>
<tr>
<td>1.5</td>
<td>23,400</td>
<td>21,437</td>
</tr>
<tr>
<td>2</td>
<td>24,457</td>
<td>24,187</td>
</tr>
</tbody>
</table>

Note: The cancellation probability was 0.1 and the overbooking penalty $130.

Acknowledgments
The authors thank Eduardo Messmacher for implementing the algorithms with cancellation and overbooking. They are also grateful to two anonymous referees for many useful suggestions.

References


Received: October 2000; revision received: September 2001; accepted: January 2002.