

Network Cargo Capacity Management

Tatsiana Levina, Yuri Levin, Jeff McGill, Mikhail Nediak

School of Business, Queen's University, Kingston, Ontario K7L 3N6, Canada

{tlevin@business.queensu.ca, ylevin@business.queensu.ca, jmcgill@business.queensu.ca, mnediak@business.queensu.ca}

We consider the problem faced by an airline that is flying both passengers and cargo over a network of locations on a fixed periodic schedule. Bookings for many classes of cargo shipments between origin-destination pairs in this network are made in advance, but the weight and volume of aircraft capacity available for cargo as well as the exact weight and volume of each shipment are not known at the time of booking. The problem is to control cargo accept/reject decisions to maximize expected profits while ensuring effective dispatch of accepted shipments through the network. This network stochastic dynamic control problem has very high computational complexity. We propose a linear programming and stochastic simulation-based computational method for learning approximate control policies and discuss their structural properties. The proposed method is flexible and can utilize historical booking data as well as decisions generated by default control policies.

Subject classifications: transportation: freight; dynamic programming: applications.

Area of review: Revenue Management.

History: Received January 2009; revisions received December 2009, May 2010; accepted August 2010.

1. Introduction

The air transportation industry derives a significant portion of its revenues from transporting cargo in both dedicated cargo and mixed passenger/cargo aircraft. Indeed, the International Air Transport Association (IATA) estimates that 2008 system-wide global revenues from cargo were \$61 billion versus \$421 billion from passengers (IATA 2009). For a large airline, the management of passenger capacity shares many features with the management of cargo capacity, including the need to control booking admissions over a global network of routes. However, while passenger capacity management has received significant attention from the research community in the revenue management (RM) literature, only recently has work emerged on the cargo problem. (The airline industry has certainly made progress in development of cargo RM systems, and there has been some academic interest; however, there has been nowhere near a comparable effort on researching the fundamental problems underlying cargo RM.)

One possible reason for this discrepancy is that the cargo problem is significantly more difficult from an algorithmic perspective. For example, while passenger capacity can be expressed simply as uniform, discrete units (seats), cargo capacity should be described by at least two continuous dimensions (weight and volume); thus, a fundamental subproblem for cargo is an NP-hard, multidimensional bin-packing problem. Also, for reasons discussed in §2, the exact cargo capacity that will be available at the time of flight departure is unknown at the time of booking, hence cargo capacity must be viewed as a random quantity until shortly before flight departure. Finally, routing and scheduling for cargo are more flexible than those for

passengers (c.f. Kasilingam 1996, p. 38). This flexibility relaxes some of the constraints on cargo capacity allocation but greatly expands the solution space that must be searched in an already complex problem, particularly when network and stochastic elements are considered.

There is extensive prior work on the passenger RM problem, and this serves as a useful reference for potential developments in the cargo area. In particular, even in the simpler passenger RM case, it is widely recognized that the network version of the problem is too large for exact solution because of the computational demands of exact formulations. Existing solution approaches have primarily focused on two types of approximate controls. One is to use single-leg methods together with a process called *virtual nesting*, which maps a multiple-leg booking request onto one of several *virtual classes* for each leg involved. The request is accepted if and only if the virtual classes on each leg are open. This control mechanism was first implemented by American Airlines around 1983. The second type of control is based on the idea of *bid-prices* introduced by Smith and Penn (1988) and Simpson (1989). In this type of control, a booking request is accepted if and only if its revenue exceeds the sum of threshold prices—bid prices—of the resources necessary to satisfy it. There is a wide range of computational approaches to finding “good” bid prices. Talluri and van Ryzin (1998) have studied theoretical properties of bid-price controls and performed an asymptotic analysis. Bertsimas and Popescu (2003) examined adaptive nonadditive bid-price controls and showed that they performed better than the traditional ones. For a detailed review of other earlier work (prior to 2004) on network RM see the book by Talluri and van Ryzin (2004).

Some recent work has focused on dynamic bid-prices and highlighted connections between bid-prices and the approximate dynamic programming approach to network RM through linear programming (see Adelman 2007 and Farias and Van Roy 2006). Meissner and Strauss (2008) apply a linear programming-based approximation methodology to find dynamic inventory-sensitive bid prices for network RM in the presence of consumer choice. Simulation-based methods (stochastic gradients) have also been used to optimize bid-price controls (see Topaloglu 2008) as well as virtual nesting controls (see van Ryzin and Vulcano 2008). We employ a combination of these latest ideas for cargo RM, and discuss the refinements necessary for LP-based computation of dynamic bid prices.

Kasilingam (1996) explores similarities and differences between passenger and cargo capacity management in the airline context and discusses potential modelling approaches to the problem. Thorough coverage of the details of airline cargo management practices and proposals for RM systems can be found in Blomeyer (2006). Recent state-of-the-art in cargo practice is discussed in Becker and Dill (2007). Several papers explore a single-leg version of the problem. Huang and Hsu (2005) tackle the problem of uncertain cargo capacity in one dimension, and Luo et al. (2009) and Moussawi and Cakanyildirim (2005) study two-dimensional cargo overbooking models. The Luo et al. article considers a cost minimization objective and shows that the overbooking limit of the one-dimensional version of the problem is replaced by a curve encircling the acceptance region in the two-dimensional version. Moussawi and Cakanyildirim consider a profit optimization objective and both aggregate and detailed formulations. The aggregate formulation serves as a lower bound, and the article also provides an upper bound. Popescu et al. (2006) provide a novel show-up rate estimation method for cargo overbooking and test it on real data. Amaruchkul et al. (2007) consider a cargo booking problem in which the state of the system is represented by the vector of numbers of accepted shipments classified by their type (an approach also taken in the current paper). The exact weights and volumes of shipments are unknown at the time of booking, but their distributions are determined by the shipment type. The article develops a heuristic based on decomposition of the problem into independent weight and volume components. Several upper bounds and heuristics are compared. A continuous-time formulation more appropriate for the ocean transport setting is studied in Xiao and Yang (2010).

The network version of the cargo problem has received much less attention. Sandhu and Klabjan (2006) consider an integer programming formulation for the fleeting problem that includes an origin-destination bid-price booking control component for air cargo on a network. However, the bid-price controls discussed in the paper are static and ignore the stochastic aspect of the booking problem, and the performance of the resulting booking controls is not evaluated. Pak and Dekker (2005) discuss a bid-price

control policy for the case of two-dimensional capacity on each flight. Shipments are considered individually, the weight and volume requirements are known exactly, and each booking request explicitly specifies an itinerary that the shipment has to follow. The complexity of the algorithm proposed by the authors grows exponentially with the number of flights. Popescu (2006) also considers booking requests in the form of explicit itineraries rather than origin-destination (OD) pairs. A distinctive feature of this work is a decomposition of the problem into small and large cargo capacity allocation subproblems. In the “large cargo” component, each shipment is represented by its weight and volume individually in the state description.

The main contributions of the current paper follow.

1. We propose an operational stochastic dynamic booking/shipping control model for air cargo capacity management on a network.
2. The model captures several sources of uncertainty inherent to air cargo:
 - (a) random booking arrival process,
 - (b) uncertain capacity of flights, and
 - (c) capacity requirements of shipments which are uncertain at the time of booking.
3. The model structure explicitly captures the periodic nature of the flight schedule. This distinguishes it from existing stochastic dynamic models in network RM.
4. We develop a solution methodology that employs dynamic capacity-dependent controls and a linear programming-based approximation procedure.
5. The proposed solution procedure for approximating this linear program is flexible and can include information about past booking/shipping policies in the solution process.

In the next section, we introduce the model formulation. In §3, we discuss the challenges posed by the model, the solution approaches used to tackle them, and the structural properties of the resulting policies. We provide numerical illustrations of this solution methodology in §4 and conclude in §5. An electronic companion to this paper is available as part of the online version at <http://or.journal.informs.org>.

2. Model Description

We consider a network of locations connected by flights operated by a single company on a fixed schedule. Our concern is efficient utilization of weight and volume cargo capacity on these flights. Such capacity is used in several ways. Some of it is reserved in the form of *allotments* to long-term cargo customers. Selling capacity through allotments involves a negotiation process that is difficult to formalize and optimize in the operational manner typical of RM systems; thus, the present formulation deals with capacity remaining after allotments have been made. However, it is a common occurrence on individual flights that capacity reserved for allotment customers is not fully used

up at flight time. Prior to flight time, the extra capacity is a component of the random amount available for other uses. Some additional capacity may be consumed by priority express shipments that are not pre-booked. If an aircraft is used for passenger travel, its cargo capacity also depends on the number of passengers (their weight) and the weight and volume of their luggage. Finally, the highly uncertain capacity remaining on each flight can be used for pre-booked, nonallotment shipments. It is this last form of capacity utilization that will be the focus of this paper because it is the most amenable to control at the time of booking. The amount of capacity available for nonallotment shipments can be significant in practice. For example, Hellermann (2006) reports that a major European carrier pre-sells, on average, about one-third of total capacity in long-term contracts. However, these allotments vary significantly from route to route within a range of 0 to 70% (Hellermann 2006, p. 15). (Air cargo capacity is routinely described in terms of weight. Even though some flights are primarily constrained in volume, air cargo capacity requirements are commonly given as *dimensional weight*—the maximum of the actual cargo weight and the standard weight corresponding to the actual cargo volume. The airlines are aware of the restrictive nature of such uniform capacity specification and seek shipments with complementary characteristics when appropriate.)

We assume that the company receives frequent nonallotment booking requests, and we model this request stream as a discrete-time counting (Bernoulli) process. The time index t of the problem ranges in the set of nonnegative integers $\mathbb{Z}_+ = \{0, 1, 2, \dots\}$. Such a booking request arrival model is appropriate if the time unit is sufficiently small that the chance of more than one request arrival in a time interval can be disregarded. Each request specifies shipment type, origin, check-in time, destination, and due time. We will refer to all this information collectively as the *booking class* (or *class* for short). The exact weight and volume of each shipment are uncertain at the time of booking, but its shipment type is known and determines the joint probability distribution of shipment volume and weight. This uncertainty arises because the physical characteristics of shipments cannot be measured precisely until their *check-in*—the time when the shipments become available for shipping. In practice, bookings can be made several weeks before the expected check-in. Shipment types were used by Amaruchkul et al. (2007) and correspond to combinations of volume and weight distribution, type of packaging, type of cargo, time sensitivity, and possibly other characteristics. Different volume and weight distributions may arise from a cargo classification scheme based on resource utilization patterns. The shipment type distinguishes among cargo variants such as fresh meat, flowers, live animals, electronic components, and so on. Shipments of one type are assumed similar across different customers, so that minor individual variations in characteristics within type can be handled by randomization away from a generic

type. Cargo of certain customers may be assigned special shipment types for “preferred” treatment. We assume that the booking arrival probability for each time interval and the probabilities that a booking request at a given time is of a particular class can be estimated from historical data. The shipments that have already physically checked-in to the system and are waiting for shipping are also handled through booking classes. From this point on, we use the term *type* exclusively as shorthand for “shipment type.”

When making accept/reject decisions for arriving booking requests, it is important to appropriately model revenues and costs associated with each request. The amount charged for a shipment depends on its booking class as well as the exact volume and weight and cannot be confirmed until the shipment is checked-in, because the last two characteristics are known only up to their joint distribution at the time of booking. However, we assume that the expected shipping charge (price) and the expected cost for each class can both be estimated. Therefore, the company knows the expected gross profit for each accepted shipment as a function of its booking class. Gross profit does not include any penalties that may be incurred in the case of failed on-time delivery. This penalty may include specific compensation to the customer, goodwill cost, and the cost of outsourcing the shipment and is charged as soon as it is detected that on-time delivery is impossible.

Time intervals can be either *departure intervals*, in which a departure is scheduled anywhere in the flight network, or *nondeparture intervals*. We assume that at the beginning of any departure interval, shipping decisions have been made for all shipments on-hand at the departure point. A booking request may arrive during a departure or nondeparture interval, in which case the company must decide whether or not to accept it. In the case of departure intervals, new booking requests are assumed to arrive after the departure. (In reality, there is a latest time that new shipments can be considered for shipping on a particular flight. Such times, rather than the actual departure times, can be used in this model.)

The company’s problem is to maximize the expected present value of profit by controlling accept/reject decisions for nonallotment booking requests and shipping accepted shipments through the network. In this problem, we assume that demand and capacity availability patterns are identical for different schedule cycles. For the sake of simplicity, we also assume the following:

- There are no additional shipping restrictions for shipments except those resulting from volume, weight, and time constraints. (For example, we exclude the possibility of jointly “incompatible” shipment types.)
- The penalty is a function of shipment type only, and once the penalty is paid, the shipment is considered “outsourced” in the model. That is, it is no longer handled by the shipping control system.

Fortunately, cargo loading restrictions and a more general cost structure can be introduced into the problem without altering the proposed modelling and solution

methodology. An electronic companion to this paper is available as part of the online version that can be found at <http://or.journal.informs.org/>. In online appendix §EC.3, we show how to relax these simplifying assumptions.

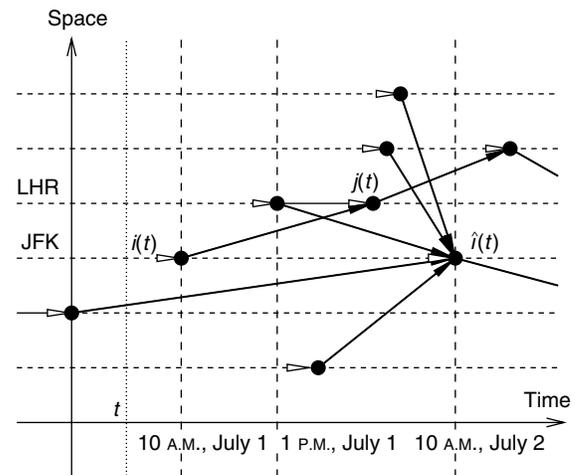
The rest of this section is organized as follows. Section 2.1 discusses the network and classification of prospective shipments according to their shipment types and booking classes; §2.2 describes the decisions, the state of the system and its evolution over time; §2.3 introduces notation related to cost/benefit terms in the objective and resource consumption/availability; §2.4 formally states the optimization model; and finally, §2.5 discusses the periodic structure of the model, solution existence, and the structure of the booking control policy.

2.1. The Network and Booking Classes

The network structure in this model reflects both spatial and temporal features of the flight schedule in a manner common in transportation network modeling. The nodes in the network correspond to flight departures and are described by both the time and location of departure. For example, there could be a node for a flight from John F. Kennedy International Airport to London Heathrow Airport (JFK-LHR) departing at 10:00 A.M. on July 1, a separate node for a LHR-JFK flight departing at 1:00 P.M. on July 1, and another node for a JFK-LHR flight departing at 10:00 A.M. on July 2. There are two kinds of arcs in the network. A *flight arc* corresponds to a flight, and for each node there is exactly one outgoing arc of this kind. (Thus, there is a one-to-one correspondence between nodes and flight arcs.) A *hold arc* corresponds to a delay of shipment until the next departure from the same location. Thus, there is exactly one outgoing hold arc from each node. Incoming arcs to a node correspond to all those flights at this location that arrive in time for transfer of their cargo to the departing flight. There is also one incoming hold arc representing a delay from the previous departure from the same location. For example, suppose there are five of the airline's flights arriving at JFK between 10:00 A.M. on July 1 and 10:00 A.M. on July 2—which are the departure times of the only two JFK-LHR flights in that time period. Label the second of these flights \hat{i} , and suppose that for all incoming flights there is enough time to transfer the cargo to this flight. Then \hat{i} is the end-node for the five incoming flights plus one hold arc from the node of the July 1st flight. Figure 1 presents a sample fragment of the flight network. Flight arcs are shown with solid lines and hold arcs with dashed lines.

Over an unlimited time horizon, the resulting network has an infinite number of arcs and nodes. However, because the schedule of flights (routes and locations) follows a periodic structure, the control policies used during current and future schedule cycles can easily be shown to be identical. We make the periodic structure explicit in our mathematical formulation later.

Figure 1. Functional representation of the spatiotemporal network structure.



The notation for a network problem of this complexity is intricate. We build the model step-by-step below and provide a summary of all notation in an appendix. Formally, the time/space network structure is described as follows. Let $I = \{1, 2, \dots\}$ be the set of nodes. For simplicity, we assume that within any given time interval there may be only one departing flight anywhere in the network. That is, any two departures are at least one time interval apart. (If there are simultaneous departures, we can always separate them by a dummy time interval in which no departures can occur.) The nodes are indexed consecutively in time. That is, $i + 1$ is the departure node of the next flight after departure i . We use the following notation:

- The number of flights in each schedule cycle is i_0 .
- The function of time $i(t): \mathbb{Z}_+ \rightarrow I$ specifies the departure node of the next flight after time t (with t strictly preceding the departure time).
- The function $j(t): \mathbb{Z}_+ \rightarrow I$ specifies the destination of the next flight after time t ; thus, the arc $(i(t), j(t))$ specifies the *origin* and *destination* of the next flight.
- The time of departure i is given by the function $\theta(i): I \rightarrow \mathbb{Z}_+$. The departure time of the next flight is then obtained as $\theta(i(t))$.
- Finally, to describe the spatial correspondence of nodes we use the function of time $\hat{i}(t): \mathbb{Z}_+ \rightarrow I$ that gives the departure node of the flight departing after $i(t)$ from the same geographic location.

Figure 1 illustrates these functions in the context of the previous JFK/LHR example.

The shipment type is denoted by $k = 1, \dots, K$, where K is the total number of types. As mentioned before, the joint probability distribution of weight and volume of a shipment is determined by its type. The booking class is a triple $q = (ijk)$, which specifies type k , the origin node $i \in I$, and the destination node $j \in I$. Because the nodes are time-location pairs, i corresponds to the unique geographic origin of a shipment and its check-in time. Similarly, j specifies

both the destination location and the due time. Without loss of generality, we assume that due times coincide with time instances described by nodes in I (departure times of flights). Because of the periodicity of the flight schedule, demand, and capacity patterns, there is a correspondence between classes over different schedule cycles. For each class $q = (ijk)$ with an origin after i_0 , the corresponding class in the previous schedule cycle is given by $\hat{q}(q) = ((i - i_0)(j - i_0)k)$.

For ease of notation in the mathematical formulation, the model encompasses all possible combinations of i , j , and k such that there is a path from i to j . In the implementation, however, this set can be significantly reduced by including only feasible arcs (i, j) in which i can be used as a first flight arc on a path from i to j . We discuss this reduction in detail in §2.2 when we explain how the system state changes as a result of flight departures.

The set of classes that might be open for booking or that arise as a result of shipping decisions during the first schedule cycle plays a special role in the model. According to usual air cargo practices, bookings are accepted no earlier than a few weeks in advance, thus classes may be open only if their corresponding departures are not too far in the future. We also need to consider other, closed classes because a shipment en route at an intermediate location does not necessarily correspond to any class for which initial bookings are being accepted. Let the set of these open or closed classes from the first cycle classes be Q , the number of elements in it be $|Q|$, and D be the smallest number of schedule cycles, such that a destination of any class in Q has the index less than or equal to $i_0 D$.

2.2. Decisions, the System State, and Its Evolution

Let λ_t equal the probability that a single booking request arrives during time interval $[t, t + 1)$. Given an arrival at t , the conditional probability that a booking request that has arrived at time t is of class q is given by π_{tq} . The state variable n_q is the current number of bookings of class q . The vector $\mathbf{n} = (n_q)$ of all such counters forms the state description of the system. While formally it is infinite-dimensional (because classes include departure and arrival nodes over an infinite horizon), only a finite subset of elements of \mathbf{n} needs to be considered at any given time. Indeed, its components are equal to zero for all classes $q = (ijk)$ for which the booking process has not yet commenced, or the flight corresponding to i has already departed. We point out that during the first schedule cycle, nonzero components of \mathbf{n} all correspond to classes in Q . To map the states corresponding to consecutive schedule cycles, we define the vector $\hat{\mathbf{n}}(\mathbf{n})$ as a state vector such that $\hat{n}_{\hat{q}(q)} = n_q$ for all $q = (ijk)$ such that $i > i_0$ and the rest of the components of $\hat{\mathbf{n}}$ are zeros. (That is, $\hat{\mathbf{n}}(\mathbf{n})$ is a vector \mathbf{n} shifted by a single schedule cycle.)

We divide decisions into two groups, with the first representing acceptance of booking requests and the second the choice of flights.

Booking decisions: The accept/reject decision for a given booking request is represented by a variable $u \in \{0, 1\}$, where $u = 1$ stands for acceptance and $u = 0$ for rejection. The post-decision number of bookings can be represented as $\mathbf{n} + u\mathbf{e}_q$, where \mathbf{e}_q is a vector with 1 in the q th position and 0 elsewhere. The decision may generally depend on time t , booking class q , and the current state \mathbf{n} . Therefore, the acceptance policy is represented by $u_{tq}(\mathbf{n})$.

Shipping decisions: The second group of decisions are those made at departure time. We recall that the flights in our model are ordered in time and there are no flights that depart at exactly the same time. A shipment leaving on flight i , has an eventual shipment destination, j' , which could require several different flights. This is distinguished from the terminal destination of flight i , which we label j . Consider flight i with destination j , and let the next flight departing from the same geographic location as i be \hat{i} . We define the variables $x_{ql} \in \{0, 1\}$, $l = 1, \dots, n_q$ as individual shipping decisions for each of the $n_q > 0$ shipments of booking class $q = (ij'k)$ available for shipping by flight i . These variables are defined for all those classes whose origin node is i and which can reach their destination on time if loaded on flight i (that is, there is a path from flight destination node j to shipment destination node j'). Let $\mathbf{x}^i = ((x_{ql}, l = 1, \dots, n_q), q = (ij'k): n_q > 0, \exists \text{ path}(j, j'))$ represent the vector of all shipping decisions for flight i . Given the initial state \mathbf{n} , we denote the state vector resulting from shipping as $\tilde{\mathbf{n}} = (\tilde{n}_{ij'k})$. To describe its components, we consider an arbitrary shipment destination j' and a type k . Given i, \hat{i}, j (determined by the flight) and j', k (determined by the booking class), there are the following groups of components of the new state vector. First, for all unaffected origin nodes $i' \neq j, \hat{i}$, we have $\tilde{n}_{(i'j'k)} = n_{(i'j'k)}$. At the flight's destination node j , the numbers of shipments for each class increase by the number shipped, while at the origin's next flight node \hat{i} , these numbers increase by the number not shipped. (The second case also applies to those shipments that cannot reach their final destination from j and do not have associated decision variables.) In both cases, the relation applies only to those shipments that can still arrive on time:

$$\begin{aligned} \tilde{n}_{(j'k)} &= n_{(j'k)} + \sum_l x_{(ij'k)l}, & \text{if } \exists \text{ path}(j, j'), \\ \tilde{n}_{(\hat{i}j'k)} &= n_{(\hat{i}j'k)} + n_{(ij'k)} - \sum_l x_{(ij'k)l}, & \text{if } \exists \text{ path}(j, j'), \text{ path}(\hat{i}, j'), \\ \tilde{n}_{(\hat{i}j'k)} &= n_{(\hat{i}j'k)} + n_{(ij'k)}, & \text{if } \nexists \text{ path}(j, j'), \exists \text{ path}(\hat{i}, j'). \end{aligned}$$

All other components of the new state vector become 0. (This is consistent with our earlier assumption that we consider outsourced any shipment that is delayed so that its

on-time delivery is impossible.) To emphasize dependence on the current state and decisions, we express the new state vector as $\tilde{\mathbf{n}}(\mathbf{n}, \mathbf{x}^i)$.

2.3. Costs, Revenues, Resource Utilization, and Availability

The objective function of the model reflects an infinite planning horizon. Future profits are discounted by factor α per time period. For a class q booking accepted at time t , the expected present value of gross profit is p_{tq} , defined as the expected discounted shipping charge minus expected discounted shipping costs (excluding possible penalties for failure to deliver on time). In practice, the revenue less the variable costs per shipment (*margin* or *contribution margin* of a shipment) is usually known for a given shipment volume and weight. (A detailed discussion of margin management principles can be found in Slager and Kapteijns 2003.) Thus, for the mathematical formulation of the problem, we assume that these values are given. Other inputs are:

- C_k —the penalty for failing to deliver a type k shipment on time,
- Y^i and Z^i —(random) volume and weight capacities of flight i , and
- V_k and W_k —(random) volume and weight of a type k shipment.

(We assume that the values of penalties as well as the distributions of the volume/weight random variables are given.) To quantify the expected present values of profit in different information states, we use the following:

- The value $J^i(t, \mathbf{n})$ denotes the expected present value of profit in state \mathbf{n} at time t after departure $i - 1$ but before departure i .

• Recall that $\theta(i)$ is the time of departure i . $J^i(\theta(i), \mathbf{n})$ refers to the expected present value of profit at the time of departure i immediately before the realizations of weights and volumes are observed, and $J^{i+1}(\theta(i), \mathbf{n})$ refers to the expected present value of profit immediately after departure i but before booking requests for period $\theta(i)$ have arrived.

• Finally, $\Phi^i(\mathbf{n}, \mathbf{v}^i, \mathbf{w}^i, y, z)$ is the expected present value of profit in state \mathbf{n} at the time of departure from i given the volumes \mathbf{v}^i and weights \mathbf{w}^i of all shipments available for shipping on flight i with realized volume and weight capacities y and z , respectively. For given i and \mathbf{n} , \mathbf{v}^i and \mathbf{w}^i include weights and volumes of all shipments with classes of the form (ijk) for some j and k .

2.4. The Optimization Model

Recursive relations for the expected present value of profit include the *booking control subproblem*

$$J^i(t, \mathbf{n}) = \lambda_t \sum_q \pi_{tq} \max_{u \in \{0,1\}} \{u p_{tq} + \alpha J^i(t+1, \mathbf{n} + u \mathbf{e}_q)\} + (1 - \lambda_t) \alpha J^i(t+1, \mathbf{n}), \quad \text{if } \theta(i-1) \leq t \leq \theta(i) - 1, \quad (1)$$

$$J^i(\theta(i), \mathbf{n}) = E_{v^i, w^i, y^i, z^i} \{\Phi^i(\mathbf{n}, V^i, W^i, Y^i, Z^i)\}, \quad i \in I, \quad (2)$$

where the terminal conditions (2) are obtained using the *shipping control subproblem* for each $i \in I$:

$$\Phi^i(\mathbf{n}, \mathbf{v}^i, \mathbf{w}^i, y, z) = \max_{\mathbf{x}^i} J^{i+1}(\theta(i), \tilde{\mathbf{n}}(\mathbf{n}, \mathbf{x}^i)) - \sum_{j: \exists \text{ path}(\hat{i}, j)} \sum_k \left(n_{(ijk)} - \sum_l x_{(ijk)l} \right) C_k, \quad (3)$$

$$\text{s.t.} \quad \sum_{jkl} x_{(ijk)l} v_{(ijk)l} \leq y, \quad (4)$$

$$\sum_{jkl} x_{(ijk)l} w_{(ijk)l} \leq z, \quad (5)$$

$$x_{(ijk)l} \in \{0, 1\}. \quad (6)$$

Equation (1) represents the expectation over arrival (with probability λ_t) or nonarrival (with probability $1 - \lambda_t$) of a booking request in the current time period and over all possible realizations of the class of this request (with probability π_{tq}). The quantity $\max_{u \in \{0,1\}} \{u p_{tq} + \alpha J^i(t+1, \mathbf{n} + u \mathbf{e}_q)\}$ gives the best expected present value of profit out of two choices: rejection ($u = 0$) or acceptance ($u = 1$) of the booking request. Equation (2) gives the terminal condition for calculation of $J^i(t, \mathbf{n})$ and reflects our assumption that in the time intervals where they occur, flight departures happen before any booking request arrivals. The vector $\mathbf{x}^i = (x_{(ijk)l})$ includes shipping decisions for all shipment destinations $j \in I$ that are reachable from the destination of flight i , all shipment types k , and all $l = 1, \dots, n_{(ijk)}$ shipments of the corresponding class (ijk) . The objective (3) of the shipping control subproblem describes the expected profit in state $\tilde{\mathbf{n}}(\mathbf{n}, \mathbf{x}^i)$ following the flight departure minus the penalty from all late shipments. The penalty is applied to any shipment such that a delay makes its on-time delivery infeasible. Thus, the first summation is over all j such that the shipments with destination j cannot be delayed until the next flight from the same location \hat{i} . This condition is expressed by the absence of the path from \hat{i} to j . Constraints (3)–(4) describe the available volume and weight.

2.5. Periodic Structure of the Model and Solution Existence

The model has a periodic structure corresponding to periodicity of the flight schedule (with a period of, for example, one week). Recall that the schedule repeats itself every i_0 flights. That is, flight 1 in the first schedule cycle corresponds to flight $i_0 + 1$ in the next schedule cycle, etc. A key observation is that only shipments of open classes or shipments en route to their final destinations can appear in the system during the first schedule cycle, and all such shipments belong to classes in the set Q . Therefore, only the state components of classes in Q may become nonzero during the first schedule cycle, and only they need to be explicitly considered in a periodic problem formulation.

LEMMA 1. We need only to consider $\mathbf{n} \in \mathbb{Z}_+^{|Q|}$ (the space of nonnegative integers indexed by the elements of Q), and the formulation (1)–(5) becomes explicitly periodic if we set

$$J^{i_0+1}(\theta(i_0), \mathbf{n}) = J^1(0, \hat{\mathbf{n}}(\mathbf{n})) \quad (7)$$

for all $\mathbf{n} \in \mathbb{Z}_+^{|Q|}$. The value of $|Q|$ is bounded in terms of the number of departures per cycle i_0 as $|Q| = O((i_0 D)^2)$.

The second statement of the lemma holds because there are at most $\frac{1}{2}(i_0 D)(i_0 D + 1)$ possible origin-destination combinations to be considered. (Recall that “big-oh” notation $O((i_0 D)^2)$ means “an expression bounded by a polynomial whose highest degree term is $(i_0 D)^2$.”) However, this is hardly comforting. For example, for an airline with 1,000 departures per week and all arcs closed within three weeks, this bound is of the order of 10 million! Clearly, heuristic approaches are required.

Because the maximum number of accepted shipments of each class in this finite-dimensional periodic model is bounded (because of the finite number of time intervals in which an arrival can occur), the state space is, in fact, finite. For such models, there exists a solution to the problem, and the method of value iteration is easily seen to converge using standard results from the theory of MDPs (see Puterman 1994). This, of course, is convergence *in principle* because no existing computer could solve a realistic problem in time for the solution to be useful.

We immediately obtain the following proposition.

PROPOSITION 1. For all $i \in I$ and $t \in \{\theta(i - 1), \dots, \theta(i) - 1\}$, there exists an optimal booking control policy of the threshold form

$$u_{i_q}^*(\mathbf{n}) = \begin{cases} 1, & \text{if } p_{i_q} \geq \alpha(J^i(t+1, \mathbf{n}) - J^i(t+1, \mathbf{n} + \mathbf{e}_q)), \\ 0, & \text{otherwise.} \end{cases} \quad (8)$$

This form of optimal policy is well known in revenue management and other problems. The formal proof is omitted because it is routine; however, it is easy to see that the proposition holds because the decision $u = 0$ results in expected present value of profit $\alpha J^i(t+1, \mathbf{n})$ while the decision $u = 1$ results in $p_{i_q} + \alpha J^i(t+1, \mathbf{n} + \mathbf{e}_q)$.

Another immediate observation from the formulation is that the optimal expected profit is decreasing in C_k , $k = 1, \dots, K$ (because reduction of C_k is essentially a relaxation of the problem). Finally, there is a straightforward equality relation between value functions at the time of departure if there is nothing to ship:

$$J^i(\theta(i), \mathbf{n}) = J^{i+1}(\theta(i), \mathbf{n})$$

for all $i = 1, \dots, i_0$, $\mathbf{n}: n_{(ijk)} = 0$ for all j, k . (9)

This observation turns out to be useful in the implementation, discussed in §3.

3. Solution Approaches

The formulation (1)–(5) poses several challenges:

- the high dimensionality of the state vector \mathbf{n} ,
- the need for a fine time discretization because of the high intensity of the booking arrival process across the network,
- the infinite periodic structure of the problem,
- the significant nonlinearity of $J^i(t, \mathbf{n})$ and $E\{\Phi^i(\mathbf{n}, V^i, W^i, Y^i, Z^i)\}$, and
- the complexity of the resulting optimal policy.

As a result of these complexities, the formulation cannot be solved exactly and has to be approximated. Generally, finding even approximate solutions to multidimensional dynamic optimization problems is not a trivial matter. In fact, it is the subject of a large and rapidly growing body of research on approximate dynamic programming (ADP), see Powell (2007). General solution ideas have to be carefully crafted to specific problems.

One promising approximation method for large-scale dynamic programming (DP) problems is based on their re-formulation as linear programs (LP). For recent general treatments see de Farias and Van Roy (2004), and for network passenger RM see Adelman (2007) and Farias and Van Roy (2006). However, the full structure of the resulting approximating LPs generally includes a constraint for each time-state-decision combination. For realistic size problems, such LPs cannot be solved exactly and require the use of approximate solution methods for LP. For example, Adelman (2007) studies the dual formulation of such an approximating LP and proposes a column-generation method to solve it (which is equivalent to an iterative constraint generation for the original problem).

In this article, we discuss an approximation method based on LP and present an iterative process for solving the approximating LP as a learning method based on the following considerations.

- Companies keep large amounts of data on past booking patterns. These data can be translated into a history of the system states, \mathbf{n} , over time, which also reflect past policies that have been considered reasonable by decision-makers. Because companies are understandably cautious about making abrupt changes to their policies, we would like to ensure that the new policy is “conservative” and has good performance on prior “representative” states.
- Companies might have some default booking policies or “rules of thumb” in place. A default policy can produce a vector of accept/reject decisions \mathbf{u} for each state in the history, and we would like to ensure a favorable comparison of any new policy with these default decisions.
- A policy improvement process that directly uses past state history and existing policies as a foundation and can incorporate new data in an adaptive on-line mode would be attractive from a practical point of view.

We incorporate these elements into a constraint sampling algorithm for solving the approximating LP. The algorithm alternates the stages of finding a control policy from

the approximating LP with a current constraint sample, and simulation of a perturbed version of this policy. The approach is based on two components: a restatement of the formulation (1)–(5) as an equivalent LP and a careful choice of a class of approximations to the value function so that the number of variables in the approximating LP is not too high. We discuss appropriate approximations to the value function in §3.1, and we state an approximating linear program in §3.2. The analysis of the approximating linear program, challenges of the LP-based approach, and details of the learning method are presented in on-line appendix §EC.2.

3.1. Approximations to the Value Function

LP-based methods and other ADP methods are often based on a particular choice of a class of approximations to the value function. Such a class is often called an *approximation architecture* in the literature (see Bertsekas and Tsitsiklis 1996).

Linear Architecture. The most popular approximation architecture—employed, for example, by Adelman (2007)—has a form that is linear in the components of the state space vector:

$$J^i(t, \mathbf{n}) \approx \alpha^{-t} F^i(t, \mathbf{n}) = \alpha^{-t} f_{i0}^i + \sum_q \alpha^{-t} f_{iq}^i n_q, \quad (10)$$

where f_{i0}^i and f_{iq}^i s are parameters of the linear approximation (to be determined) measured in terms of the present value at time 0 ($F^i(t, \mathbf{n})$ is converted to the present value at time t by the factor α^{-t}). An attractive feature of this approximation for the air cargo problem is that the booking control subproblem of the formulation admits an exact solution. However, this solution does not fully capture either the temporal structure of the arrival process or the dependencies between different classes. Specifically, the following proposition holds (proof in the appendix):

PROPOSITION 2. *Suppose we substitute approximation (10) at time $t + 1$ into (1). Then, the resulting heuristic policy is given by*

$$u_{iq}^L(\mathbf{n}) = \begin{cases} 1, & \text{if } \alpha^t p_{iq} + f_{(t+1)q}^i \geq 0, \\ 0, & \text{otherwise,} \end{cases} \quad (11)$$

and the approximate value function computed according to (1) after substitution also belongs to class (10). Moreover, the coefficients of this approximation are given by

$$f_{iq}^i = f_{(t+1)q}^i \quad (12)$$

for all q , with a constant term of the form

$$f_{i0}^i = f_{(t+1)0}^i + \lambda_t \sum_q \pi_{iq} \{ \alpha^t p_{iq} + f_{(t+1)q}^i \}^+, \quad (13)$$

where $\{a\}^+ = \max\{a, 0\}$ denotes the positive part of a .

While acceptance rule (11) appears to be dynamic because (formally) we allow f_{iq}^i s to vary with time, the form of solution (12) indicates that the policy is actually a static one, as long as the present value $\alpha^t p_{iq}$ of gross profit at time 0 is constant with respect to t .

Despite this drawback, linear approximation could be used for implementation of a Lagrangian relaxation bound for shipping control. If we relax constraints (4)–(5) while adding the corresponding Lagrangian terms to the objective function, the resulting Lagrangian is linear in \mathbf{x} , and for each realization of weights and volumes $\mathbf{v}^i, \mathbf{w}^i, y, z$, there is a closed-form integer solution. The expected value of this Lagrangian bound can, in turn, be bounded. However, our numerical tests show that the overall bound on expected profits is very loose, and the resulting acceptance policy is rather poor. The primary reason for this is a failure of the linear approximation to adequately represent the load of the system at a particular state and to capture dependencies between booking classes. The first-order information can still be captured because the expected profits should generally be decreasing in each class, and this effect *can* be represented by a linear approximation. However, the resulting decision rule (11) is state-independent. As a result, a decision to accept or reject a particular class does not depend on the number of accepted shipments from this and other classes. Thus, the linear approximation is unlikely to perform well when used globally (with the same coefficients for all \mathbf{n}).

Quadratic Architecture. The limitations of linear architecture prompted us to study an approximation architecture that is linear in the parameters but quadratic in the components of the state space:

$$J^i(t, \mathbf{n}) \approx \alpha^{-t} F^i(t, \mathbf{n}) = \alpha^{-t} f_{i0}^i + \sum_q \alpha^{-t} f_{iq}^i n_q + \sum_{(\beta, \gamma) \in S} \alpha^{-t} f_{i\beta\gamma}^i (\mathbf{d}_\beta^\top \mathbf{n})(\mathbf{d}_\gamma^\top \mathbf{n}), \quad (14)$$

where $f_{i\cdot}^i$ s are parameters of the approximation and \mathbf{d} s are fixed vectors of the same dimension as the state space. The collection of vectors $\mathbf{d}_\gamma \geq 0, \gamma \in \Gamma$ must be selected to reflect the most important features of the state space. In practice, these features should incorporate the intuition of the cargo manager about important aspects of the cargo load on each flight that need to be taken into account when making a booking decision. For example, the total expected weight or volume of bookings for a particular flight is important for booking/shipping decisions for shipments utilizing this flight. The set of pairs $S \subseteq \Gamma \times \Gamma$ should be chosen to reflect such dependencies. For example, decisions affecting the total expected weight of shipments scheduled for a flight also depend on the total expected weight of shipments that will be added to it by a connecting flight. In practice, many of these dependencies between features follow the topology of the network and can be identified by analyzing the shipments on flights with the highest utilization.

Such an approximation has the advantage of more accurately representing the value function, but it is harder to design numerical procedures for it, since it is not separable in the components of the state space. While there are applications of separable nonlinear architectures (for example, Godfrey and Powell 2002), nonseparable quadratic architectures are rarely used. In fact, we are not aware of any works using a nonseparable quadratic architecture except that of Marbach et al. (2000), who study call admission and routing in a telecommunication network. That problem is similar to ours in its network aspect, but the computational approach cannot be directly applied here because the processes of routing a call and a shipment are significantly different. In particular, the structure of their network does not involve time, calls tie-up the capacities of the links they are routed through for their entire duration, and the capacity becomes available once a call is finished.

The acceptance policy resulting from the quadratic approximation is described by the following proposition.

PROPOSITION 3. *Suppose we substitute approximation (14) at time $t + 1$ into (1). Then, the resulting heuristic policy is given by*

$$u_{iq}^F(\mathbf{n}) = \begin{cases} 1, & \text{if } \alpha^t p_{iq} + f_{(t+1)q}^i + \sum_{(\beta, \gamma) \in S} f_{(t+1)\beta\gamma}^i (d_{\beta q} d_{\gamma q} \\ & + d_{\gamma q} \mathbf{d}_{\beta}^T \mathbf{n} + d_{\beta q} \mathbf{d}_{\gamma}^T \mathbf{n}) \geq 0, \\ 0, & \text{otherwise.} \end{cases} \quad (15)$$

The quantity

$$-f_{(t+1)q}^i - \sum_{(\beta, \gamma) \in S} f_{(t+1)\beta\gamma}^i (d_{\beta q} d_{\gamma q} + d_{\gamma q} \mathbf{d}_{\beta}^T \mathbf{n} + d_{\beta q} \mathbf{d}_{\gamma}^T \mathbf{n})$$

can be viewed as a bid price for the average resources consumed by a shipment of booking class q , and the whole expression in (15) corresponding to $u = 1$ (acceptance) is the present value at time 0 of the marginal return on a booking of class q (as estimated by the approximation). These quantities have natural economic properties if we restrict the values of all second-order coefficients so that $F^i(t, \mathbf{n})$ is submodular and directionally concave in \mathbf{n} for $\mathbf{n} \geq 0$. Submodularity and directional concavity of the value function correspond to the property of diminishing marginal returns for an extra booking of each class in the number of bookings of all classes. Such a property is consistent with economic intuition. This is summarized in the following proposition.

PROPOSITION 4. *Suppose that for each i , the approximation $F^i(t, \mathbf{n})$ is submodular and directionally concave in \mathbf{n} for $\mathbf{n} \geq 0$. Then the following statements hold:*

(a) *As estimated by the approximation, the present value at time 0 of marginal return on a booking of every class is decreasing in each component of the state vector \mathbf{n} .*

(b) *For each class q , given state components \mathbf{n}_{-q} of all other classes, the maximum value of n_q for which new class q shipments are accepted is*

$$\bar{n}_{iq}^i(\mathbf{n}_{-q}) = \begin{cases} 0, & \text{if } \alpha^t p_{iq} + g_{iq0}^i + (\mathbf{g}_{iq(-q)}^i)^T \mathbf{n}_{-q} \leq 0 \text{ and } g_{iqq}^i = 0 \\ +\infty, & \text{if } \alpha^t p_{iq} + g_{iq0}^i + (\mathbf{g}_{iq(-q)}^i)^T \mathbf{n}_{-q} > 0 \text{ and } g_{iqq}^i = 0 \\ \left[-\frac{1}{g_{iqq}^i} (\alpha^t p_{iq} + g_{iq0}^i + (\mathbf{g}_{iq(-q)}^i)^T \mathbf{n}_{-q}) \right], & \text{if } g_{iqq}^i < 0 \end{cases} \quad (16)$$

where

$$g_{iq0}^i = f_{(t+1)q}^i + \sum_{(\beta, \gamma) \in S} f_{(t+1)\beta\gamma}^i d_{\beta q} d_{\gamma q}, \quad (17)$$

$$\mathbf{g}_{iq}^i = \sum_{(\beta, \gamma) \in S} f_{(t+1)\beta\gamma}^i (d_{\gamma q} \mathbf{d}_{\beta} + d_{\beta q} \mathbf{d}_{\gamma}), \quad (18)$$

and $\mathbf{g}_{iq(-q)}^i$ are all components of \mathbf{g}_{iq}^i except for n_q . Moreover, the threshold $\bar{n}_{iq}^i(\mathbf{n}_{-q})$ is nonincreasing in each component of \mathbf{n}_{-q} .

Part (a) of the proposition is immediate, well known, and presented here for completeness. In part (b), we provide an expression for the acceptance thresholds of classes specific to our approximation function. The quantity $-g_{iq0}^i$ corresponds to the expected opportunity cost of a class q booking if accepted at time t in the empty system. The vector $-\mathbf{g}_{iq}^i$ gives the rates of change in class q opportunity cost per booking of each class. The requirement of submodularity and directional concavity of $F^i(t, \mathbf{n})$ can be guaranteed, for example, by nonpositivity of the second-order coefficients $f_{i\beta\gamma}^i$ together with nonnegativity of \mathbf{d}_{β} and \mathbf{d}_{γ} . Less restrictive conditions can be used under specific choices of crossproducts. One such choice is discussed in online appendix EC.2.1.

3.2. An Approximating Linear Program

The basis for the approximating LP is a restatement of formulation (1)–(5) augmented with the periodic restriction (7) as LP. Suppose that the joint distribution of $V^i, W^i, Y^i, Z^i, i \in I_0 = \{1, \dots, i_0\}$ corresponding to each possible state \mathbf{n} at the time of departure i has a finite support indexed by the elements of the set $R^i(\mathbf{n})$. Let the probability of support point $\mathbf{v}^{i\rho}, \mathbf{w}^{i\rho}, \mathbf{y}^{i\rho}, \mathbf{z}^{i\rho}$ with an index $\rho \in R^i(\mathbf{n})$ be σ_{ρ} . Also, let $b(\mathbf{n})$ be an arbitrary initial probability distribution for the system state \mathbf{n} . The problem of maximizing $\sum_{\mathbf{n}} b(\mathbf{n}) J^1(0, \mathbf{n})$ can be restated using (1)–(5), (7) as the following LP, where the value functions $J^i(\cdot)$ and $\Phi^i(\cdot)$

are replaced with the corresponding variables $\tilde{J}^i(\cdot)$ and $\tilde{\Phi}^i(\cdot)$:

$$\min \sum_{\mathbf{n}} b(\mathbf{n}) \tilde{J}^1(0, \mathbf{n}), \quad (19)$$

$$\text{s.t. } \tilde{J}^i(t, \mathbf{n}) \geq \lambda_t \sum_q \pi_{tq} \{u_q p_{tq} + \alpha \tilde{J}^i(t+1, \mathbf{n} + u_q \mathbf{e}_q)\} \\ + (1 - \lambda_t) \alpha \tilde{J}^i(t+1, \mathbf{n})$$

$$\text{for all } i \in I, t \in \{\theta(i-1), \dots, \theta(i)-1\}, \mathbf{u}, \mathbf{n}, \quad (20)$$

$$\tilde{J}^i(\theta(i), \mathbf{n}) = \sum_{\rho \in R^i(\mathbf{n})} \sigma_{\rho} \tilde{\Phi}^i(\mathbf{n}, \mathbf{v}^{i\rho}, \mathbf{w}^{i\rho}, y^{i\rho}, z^{i\rho}) \\ \text{for all } i \in I_0, \mathbf{n}, \quad (21)$$

$$\tilde{\Phi}^i(\mathbf{n}, \mathbf{v}^{i\rho}, \mathbf{w}^{i\rho}, y^{i\rho}, z^{i\rho}) \geq \tilde{J}^{i+1}(\theta(i), \tilde{\mathbf{n}}(\mathbf{n}, \mathbf{x}^i)) \\ - \sum_{j: \exists \text{ path}(i,j)} \sum_k (n_{(ijk)} - \sum_l x_{(ijk)l}^{\rho}) C_k$$

for all $i \in I_0, \mathbf{n}, \rho$, and \mathbf{x}^i , such that

$$\begin{cases} \sum_{jkl} x_{(ijk)l}^{\rho} v_{(ijk)l}^{\rho} \leq y^{i\rho} \\ \sum_{jkl} x_{(ijk)l}^{\rho} w_{(ijk)l}^{\rho} \leq z^{i\rho} \\ x_{(ijk)l}^{\rho} \in \{0, 1\} \end{cases} \quad (22)$$

$$\tilde{J}^{i_0+1}(\theta(i_0), \mathbf{n}) = \tilde{J}^1(0, \hat{\mathbf{n}}(\mathbf{n})) \quad \text{for all } \mathbf{n}. \quad (23)$$

For every feasible solution, the variables $\tilde{J}^i(\cdot)$ and $\tilde{\Phi}^i(\cdot)$ provide upper bounds for the corresponding value functions (see, e.g., Puterman 1994). Similarly, objective function (19) in this problem provides an upper bound on the present expected value of profit given that the system starts from the initial probability distribution $b(\mathbf{n})$. The constraints that are tight in the optimal solution to this problem correspond to optimal policy decisions in the original DP (as long as the corresponding state occurs with nonzero probability).

Of course, the above formulation is no more tractable than the original dynamic program because of an extremely large number of variables and constraints. The $\tilde{\Phi}^i(\mathbf{n}, \mathbf{v}^{i\rho}, \mathbf{w}^{i\rho}, y^{i\rho}, z^{i\rho})$ reward variables are actually not needed if we endow \mathbf{x}^i with an additional superscript ρ to indicate a separate constraint index. Then (20)–(22) can be restated as a single constraint

$$\tilde{J}^i(\theta(i), \mathbf{n}) \geq \sum_{\rho \in R^i(\mathbf{n})} \sigma_{\rho} \left[\tilde{J}^{i+1}(\theta(i), \tilde{\mathbf{n}}(\mathbf{n}, \mathbf{x}^{i\rho})) \right. \\ \left. - \sum_{j: \exists \text{ path}(i,j)} \sum_k \left(n_{(ijk)} - \sum_l x_{(ijk)l}^{\rho} \right) C_k \right] \\ \text{for all } i \in I_0, \mathbf{n}, \text{ and } \mathbf{x}^{i\rho}, \text{ such that}$$

$$\begin{cases} \sum_{jkl} x_{(ijk)l}^{\rho} v_{(ijk)l}^{\rho} \leq y^{i\rho} \\ \sum_{jkl} x_{(ijk)l}^{\rho} w_{(ijk)l}^{\rho} \leq z^{i\rho}, \quad \rho \in R^i(\mathbf{n}). \\ x_{(ijk)l}^{\rho} \in \{0, 1\} \end{cases} \quad (24)$$

Unfortunately, the number of $\tilde{J}^i(\cdot)$ variables is still too large. Thus, we seek to obtain yet a simpler problem by an appropriate substitution of variables representing the value function. For example, after substituting quadratic $\tilde{J}^i(t, \mathbf{n}) = \alpha^{-t} F^i(t, \mathbf{n})$ for all $t \in \{\theta(i-1), \dots, \theta(i)\}$ and all i into (19)–(20), (23), (24), we obtain

$$\min \sum_{\mathbf{n}} b(\mathbf{n}) \left(f_{00}^1 + \sum_q f_{0q}^1 n_q + \sum_{(\beta, \gamma) \in S} f_{0\beta\gamma}^1(\mathbf{d}_{\beta}^{\top} \mathbf{n})(\mathbf{d}_{\gamma}^{\top} \mathbf{n}) \right), \quad (25)$$

$$\text{s.t. } f_{t0}^i + \sum_q f_{tq}^i n_q + \sum_{(\beta, \gamma) \in S} f_{t\beta\gamma}^i(\mathbf{d}_{\beta}^{\top} \mathbf{n})(\mathbf{d}_{\gamma}^{\top} \mathbf{n}) \\ \geq f_{(t+1)0}^i + \sum_q f_{(t+1)q}^i n_q + \sum_{(\beta, \gamma) \in S} f_{(t+1)\beta\gamma}^i(\mathbf{d}_{\beta}^{\top} \mathbf{n})(\mathbf{d}_{\gamma}^{\top} \mathbf{n}) \\ + \lambda_t \sum_q \pi_{tq} u_q \left\{ \alpha^t p_{tq} + f_{(t+1)q}^i \right. \\ \left. + \sum_{(\beta, \gamma) \in S} f_{(t+1)\beta\gamma}^i (d_{\beta q} d_{\gamma q} + d_{\gamma q} \mathbf{d}_{\beta}^{\top} \mathbf{n} + d_{\beta q} \mathbf{d}_{\gamma}^{\top} \mathbf{n}) \right\} \\ \text{for all } i \in I_0, t \in \{\theta(i-1), \dots, \theta(i)-1\}, \mathbf{u}, \mathbf{n}, \quad (26)$$

$$f_{\theta(i)0}^i + \sum_q f_{\theta(i)q}^i n_q + \sum_{(\beta, \gamma) \in S} f_{\theta(i)\beta\gamma}^i(\mathbf{d}_{\beta}^{\top} \mathbf{n})(\mathbf{d}_{\gamma}^{\top} \mathbf{n}) \\ \geq \sum_{\rho \in R^i(\mathbf{n})} \sigma_{\rho} \left[f_{\theta(i)0}^{i+1} + \sum_q f_{\theta(i)q}^{i+1} \tilde{n}_q(\mathbf{n}, \mathbf{x}^{i\rho}) \right. \\ \left. + \sum_{(\beta, \gamma) \in S} f_{\theta(i)\beta\gamma}^{i+1}(\mathbf{d}_{\beta}^{\top} \tilde{\mathbf{n}}(\mathbf{n}, \mathbf{x}^{i\rho}))(\mathbf{d}_{\gamma}^{\top} \tilde{\mathbf{n}}(\mathbf{n}, \mathbf{x}^{i\rho})) \right. \\ \left. - \sum_{j: \exists \text{ path}(i,j)} \sum_k \left(n_{(ijk)} - \sum_l x_{(ijk)l}^{\rho} \right) C_k \right]$$

for all $i \in I_0, \mathbf{n}$ and $\mathbf{x}^{i\rho}$ such that

$$\begin{cases} \sum_{jkl} x_{(ijk)l}^{\rho} v_{(ijk)l}^{\rho} \leq y^{i\rho} \\ \sum_{jkl} x_{(ijk)l}^{\rho} w_{(ijk)l}^{\rho} \leq z^{i\rho}, \quad \rho \in R^i(\mathbf{n}), \\ x_{(ijk)l}^{\rho} \in \{0, 1\} \end{cases} \quad (27)$$

$$f_{\theta(i_0)0}^{i_0+1} + \sum_q f_{\theta(i_0)q}^{i_0+1} n_q + \sum_{(\beta, \gamma) \in S} f_{\theta(i_0)\beta\gamma}^{i_0+1}(\mathbf{d}_{\beta}^{\top} \mathbf{n})(\mathbf{d}_{\gamma}^{\top} \mathbf{n}) \\ = \alpha^{\theta(i_0)} \left(f_{00}^1 + \sum_q f_{0q}^1 \hat{n}_q(\mathbf{n}) + \sum_{(\beta, \gamma) \in S} f_{0\beta\gamma}^1(\mathbf{d}_{\beta}^{\top} \hat{\mathbf{n}}(\mathbf{n}))(\mathbf{d}_{\gamma}^{\top} \hat{\mathbf{n}}(\mathbf{n})) \right) \\ \text{for all } \mathbf{n}. \quad (28)$$

A few comments are in order. Although this approximating linear program appears to be complex, it inherits some

natural properties from the exact linear program formulation (19)–(23). In particular, the objective (25) corresponds to the expected present value of profit of the problem at time zero (as estimated by the quadratic approximation) given that the system state has initial distribution $b(\cdot)$. The constraint (26) states that for each interdeparture interval, each time instance, and each state-decision pair \mathbf{n}, \mathbf{u} , the difference between the present and future quadratic approximation in the same state is bounded below by

$$Pr[\text{arrival}] \times E[\text{profit} - \text{opportunity cost}]$$

in state \mathbf{n} and given that the policy decisions \mathbf{u} are used. The structure of the opportunity cost estimate is made transparent by the explicit substitution of the quadratic approximation. The constraint (27) has a similar flavor and states that, for every departure, state, and possible shipping decisions $\mathbf{x}^{i\rho}$ under realization ρ of volumes and weights, the quadratic approximation at the time just before departure i is bounded below by the expected value of the approximation right after the departure. Finally, constraint (28) expresses the stationarity of the periodic value function after appropriate discounting.

The approximating linear program (25)–(28) forms a basis for the iterative learning method described in detail in online appendix §EC.2. The key points addressed are the choice of the quadratic terms in the approximation, a reduction of the space of the approximation by choosing a specific parametric dependence of its coefficients on time, and the approximating LP constraint generation process.

4. Numerical Illustrations

In this section, we illustrate the viability of the proposed model and approximation method on two examples. The first example is sufficiently small that we can compare the quadratic approximation (QA) with the exact solution approach (ES) in the acceptance control problem (exact solution in shipping control is not practical even for a small problem). QA offers performance that is comparable to that of ES, while the resulting acceptance policy has a much simpler structure. The “tiny” size of the first sample problem necessary for it to be tractable underscores the great complexity of this problem in general. In the second example, the number of classes is higher, and it is impossible to compute optimal acceptance controls exactly. Thus, in this example, we compare a first-come-first-served (FCFS) policy, which approximates a plausible practical heuristic, with two versions of quadratic approximation—separable (QAS), and general (QA). In the QAS approximation, only terms of the form $f_{iqq}^i n_q^2$ are considered, while the QA approximation allows general cross-product terms. The results show that there are significant benefits from using the quadratic approximation rather than FCFS. However, the quadratic approximation performs better if it includes cross-product terms. In both cases, we used parallel search process with simulation-based generation of constraints for the approximating LP (25)–(28) (as described

in §EC.2.2). The initial set included the default constraints implied by the structure of the approximating LP as well as those corresponding to states occurring in FCFS policy simulations. The quadratic approximation architecture also used an explicit polynomial parametric dependence over time defined by (EC.20)–(EC.22).

4.1. Performance of Quadratic Approximation in Comparison with the Exact Solution in Acceptance Control

In this example we consider the simplest possible network in terms of space: alternating flights connecting two locations with symmetric demand patterns. Because of the symmetry in demands and the network topology, it is enough to explicitly consider only one of these locations. The schedule cycle has one departure only; however, some shipments have a sufficiently large delivery window that their shipment can be delayed if necessary. We consider three classes of shipments:

1. those requiring service in the schedule cycle in which they are booked and checked in for shipping,
 2. those requiring current or next-cycle service but booked and checked in for shipping in the current cycle, and
 3. those requiring service in the same cycle in which they are checked in for shipping but booked in advance.
- The resulting state space is three-dimensional. Any booked shipments of the second class that cannot be shipped in the current cycle are flexible and can be delayed until the next cycle in which they effectively become the shipments of booking class 1. The same treatment applies to inflexible bookings of class 3.

We compare performance of policies computed with the following solution approaches in acceptance control: first, the exact solution using dynamic programming (ES) and, second, the quadratic approximation architecture (QA) selected by the learning method. The dynamic programming approach (ES) still relies on a quadratic approximation of the value function to find an approximate solution to the shipping control subproblem. This involves fitting a quadratic model to approximate $J^1(0, \mathbf{n})$ at the beginning of the planning horizon. Only dependence of $J^1(0, \mathbf{n})$ on the number of shipments of class 1 is relevant for this model, because any unshipped shipments of class 2 and booked shipments of class 3 are equivalent to shipments of class 1 after the departure. Given the quadratic approximation, we use a heuristic maximization to approximate a shipping subproblem (as described in §EC.2.3). The results of this heuristic are then averaged over a volume/weight parameter sample. The policies are compared with simulation.

Below, we provide the detailed parameters for this example. Because the shipments do not differ in type but only in their origin and destination nodes, we drop the type subscript from our notation.

- For dynamic programming, we limit the state space to 20 shipments of each class.
- The schedule cycle is partitioned into $\theta(1) = 200$ time intervals.
- Volume V and weight W of each shipment are bivariate Normal(1, 0.5) with correlation $\rho = 0.5$.
- Weight capacity Y_i of each departure is Normal(10, 2), and volume capacity Z_i is Normal(μ , 2) with $\mu = 10, 12, 14, 16$ (we test four different values for the mean).
- The penalty for failure to deliver on time is $C = 10$.
- We test the following booking arrival rates: $\lambda = 0.05, 0.10, 0.15$.
- The total discount factor is 0.9 per cycle (that is $\alpha^{\theta(1)} = 0.9$).
- Finally, gross profits and arrival probabilities are given in the following table:

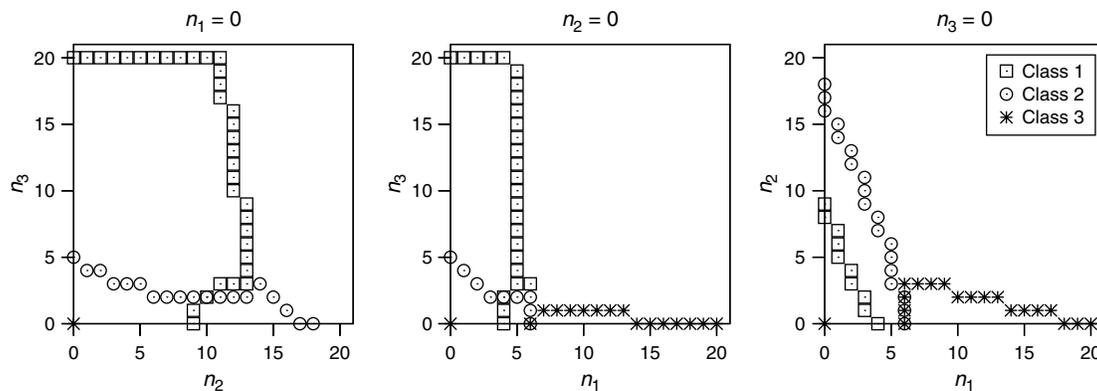
Class	1	2	3
Gross profit	2	1	1.5
π_q	0.4	0.3	0.3

The state space for the simulation is limited as discussed in §EC.2.3. The feature vectors in the quadratic approximation and their pairs S are chosen as described in §EC.2.2, and the components of the \mathbf{d}_γ s are equal to the average weight of the corresponding shipment. The quadratic and linear subproblems, and a multicommodity flow problem needed to limit the state space (described in online appendix EC.2.3) are solved using the COIN/CLP package, which is available from <http://www.coin-or.org/>. In the “exact” dynamic programming approach, the value iteration is repeated 100 steps (one step per schedule cycle). In the LP-based search procedure, we also use 100 iterations in 8 independent searches. These searches, guided by different random number sequences, are executed in parallel on a SHARCNET serial throughput cluster (Shared Hierarchical Academic Research Computing Network: <http://www.sharcnet.ca/>). Next, we discuss the results of the first experiment.

Acceptance regions obtained using the exact dynamic programming calculation in acceptance control for three classes are shown in Figure 2. Because three-dimensional plots are generally hard to interpret, we present two-dimensional slices of the acceptance regions of different classes along the subspaces $n_1 = 0$, $n_2 = 0$, and $n_3 = 0$. Three different line styles indicate the boundaries of acceptance regions for three different classes. The main features of these regions are the dependencies of acceptance decisions of all classes on the number of reservations for other classes. We may conclude that a class A acceptance decision depends strongly on the number of accepted reservations of another class, B , if acceptance for A ends abruptly when the number of reservations for B increases. Generally, acceptance decisions of classes 1 and 3 (“square” and “star” lines) strongly depend on the number of accepted reservations n_2 of class 2 (and themselves). Class 2 (“circle” line) acceptance decisions also strongly depend on n_1 and n_3 . However, the dependence between classes 1 and 3 is weaker. The shape of the acceptance regions, while complex, is characterized by long, almost linear segments. Thus, we may hope that the quadratic approximation architecture, which captures some dependencies between classes and results in linear acceptance regions, is adequate.

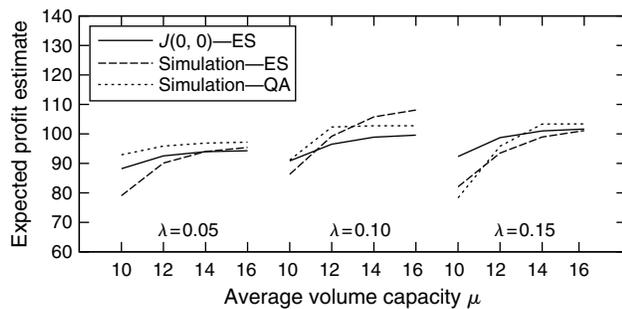
Figure 3 shows the estimates (obtained by simulation) of expected present values of profit at the beginning of the cycles for two policy computation approaches (dashed lines) as well as the expected profit $J^1(0, \mathbf{0})$ in state $\mathbf{n} = \mathbf{0}$ computed according to the ES approach (solid line). All possible experimental settings are shown. The plots are arranged in three groups according to the magnitude of the arrival rate λ . Within each group, the graphs are profits versus volume capacity $\mu \in \{10, 12, 14, 16\}$. Overall, all profit estimates increase as the volume constraint becomes less tight (μ increases). There is also a tendency toward an increase as the arrival rate λ increases. In the case of the ES approach, $J^1(0, \mathbf{0})$ may over- or under-estimate the simulated performance of the policy. The overestimation (underestimation) is associated with a tighter (looser) weight capacity constraint. The performance of QA and ES

Figure 2. Acceptance regions at the beginning of the schedule cycle when the acceptance control problem is solved exactly: two-dimensional slices over subspaces $n_1 = 0$, $n_2 = 0$, and $n_3 = 0$.



INFORMS holds copyright to this article and distributed this copy as a courtesy to the author(s). Additional information, including rights and permission policies, is available at <http://journals.informs.org/>.

Figure 3. Expected profit estimates for all experimental settings.



policies is very close, which suggests that the QA policy may be promising for larger-scale situations when the exact calculation of acceptance control is computationally infeasible. Next, we look into one such example.

4.2. Benefits of Control Policies Based on Nonseparable Quadratic Architecture

In this subsection, we consider schedules with $i_0 = 4$ or $i_0 = 8$ departures per cycle. The spatial network topology includes one central location, a hub, and peripheral locations, satellites, which do not have direct transportation links but communicate through the hub. The departures in each schedule cycle occur in the following order: the hub to spoke 1, spoke 1 to the hub, the hub to spoke 2, and spoke 2 to the hub, etc. The case of 4 departures per cycle corresponds to 2 spokes, and the case of 8 departures—to 4. Potential bookings may request travel between the hub and the spokes as well as between the spokes in either direction. Shipments are also distinguished by whether they require an immediate service after their delivery for shipping or can wait until the next schedule cycle. Bookings are accepted strictly less than 2 schedule cycles in advance resulting in open reservations for 8 and 16 flights in the cases of 4 and 8 departures per cycle, respectively. Finally, there are 2 shipment types. All possible combinations of these elements result in 48 and 160 simultaneously open classes, respectively.

In this experiment, the quadratic architecture includes linear and quadratic terms in individual state components for all classes. For more complex quadratic terms, we tested two options: none, and the terms described in §EC.2. We refer to acceptance policies resulting from quadratic architectures of these two forms as QAS (separable) and QA policies, respectively.

QA and QAS policies are compared with a variation of a first-come-first-served policy (FCFS) that takes only available capacity into account and does not rely on any RM technique. In this policy, the route to be taken is fixed upon booking, and capacity availability is determined from the remaining (unbooked) capacity of the flights. Because flight capacity is uncertain and the company would like to avoid being charged penalties for failure to deliver on time, we

use the mean flight capacity minus a safety factor as the initial unbooked capacity of the flight. We experimented with this additional safety margin around the mean to find a margin that provided the maximum expected profit. When realized capacity of the flights turns out to be insufficient, the heuristic attempts to reroute the shipments (in the case of a one-hub network, rerouting is a simple delay of shipment).

Other details of this example are as follows:

- Each interdeparture interval is partitioned into 10,000 time intervals.
- The volume and weight of each type 1 shipment are bivariate Normal(1, 0.5) with correlation $\rho = 0.5$. Type 2 differs from type 1 in its volume distribution, which is Normal(0.5, 0.25).
- Volume Y_i and weight Z_i capacity of each departure are Normal(20, σ), with $\sigma \in \{2, 5\}$ (we test two values for standard deviation).
- The penalty for failure to deliver on time is $C = 10$.
- The booking arrival rate and arrival probabilities are chosen at random to simulate nonhomogeneity: for each open booking class q , $\lambda\pi_{i,q}$ is sampled from the uniform distribution on the interval $[\lambda/2, 0.000, 3\lambda/2, 0.000]$, where we test $\lambda \in \{0.05, 0.10, 0.25, 0.5\}$.
- The total discount factor is 0.9 per cycle (that is $\alpha^{(4)} = 0.9$).
- Finally, the expected gross profit for a class is computed using the following rules: starting with the “base” value of 2.0, increase the price by 1.0 (in absolute value) if the class requires a connection, reduce it by 0.5 (in absolute value) if the class is of type 2 (consumes less volume), and reduce it by $\delta \in \{0.2, 0.5\}$ if the shipping may be delayed, or if the booking is more than one cycle in advance (all reductions are cumulative). A smaller value of δ corresponds to a harder problem from the standpoint of the booking control effectiveness because the margins of shipments are closer to each other, and thus it is harder to identify the shipments providing better results.

Eight parallel instances of the search algorithm were run for each instance.

The performance of the QA and QAS acceptance policies as a percentage difference from the present value of profit obtained by the FCFS policy is given in Table 1. The percentage difference is computed as

$$\left(\frac{\text{estimated present value of QAS or QA profit}}{\text{estimated present value of FCFS profit}} - 1 \right) \times 100\%.$$

The present values of profit for QAS or QA policies are estimated by simulating 1,000 schedule cycles, splitting the results into 100 consecutive 10-cycle runs, and averaging the estimates obtained from the 10-cycle present values (the latter are corrected for the fact that the time horizon is truncated at 10 cycles in each run). The present values for FCFS are obtained similarly from 10,000 schedule cycles split into 500 runs of 20 consecutive cycles. The first important observation from this table is that both policies, QA

Table 1. Performance of the QA and QAS policies as a percentage improvement relative to the expected present value of profit obtained by the FCFS policy for various levels of flexibility/advance booking discount, δ , capacity uncertainty, σ , and magnitudes of arrival rate, $\bar{\lambda}$.

δ	σ	$\bar{\lambda}$	$i_0 = 4$ (%)		$i_0 = 8$ (%)	
			QAS	QA	QAS	QA
0.2	2	0.05	16.7	19.6	15.2	16.5
0.2	2	0.10	25.1	22.8	14.8	17.8
0.2	2	0.25	19.8	27.1	14.4	19.6
0.2	2	0.50	20.5	29.6	13.6	19.3
0.2	5	0.05	10.7	10.8	6.2	4.2
0.2	5	0.10	11.8	12.0	4.5	5.5
0.2	5	0.25	10.4	14.1	7.5	8.6
0.2	5	0.50	8.0	12.9	10.1	12.3
0.5	2	0.05	39.4	41.9	51.0	46.6
0.5	2	0.10	74.7	63.3	50.6	48.3
0.5	2	0.25	64.2	63.4	47.3	47.0
0.5	2	0.50	60.3	62.5	46.5	47.9
0.5	5	0.05	47.4	42.1	42.8	44.8
0.5	5	0.10	71.4	58.6	39.5	44.7
0.5	5	0.25	54.4	53.8	49.9	47.8
0.5	5	0.50	49.1	57.9	53.8	49.6
Average			36.5	37.0	29.2	30.0

Note. Best performance in bold.

and QAS, always perform significantly better than FCFS. The gains of QA and QAS are generally higher in “easier” cases of a high discount for flexibility and advance booking $\delta = 0.5$ and a small level in capacity uncertainty $\sigma = 2$. We also see that the QA policy slightly outperforms QAS on average (QA’s average gain over QAS is 0.53% when $i_0 = 4$ and 0.80% when $i_0 = 8$). The advantage of QA is very clear in the harder case of a small δ (QA beats QAS in 14 out of 16 settings with $\delta = 0.2$). However, QAS performs better than QA in 10 settings out of 16 for high δ . The maximum relative performance of QAS is the best for $i_0 = 4$, $\delta = 0.5$ and an intermediate level of arrival intensity $\bar{\lambda} = 0.10$. A possible explanation is that QAS has fewer parameters while its structure still allows the learning algorithm to tune booking limits for individual shipments. Such individual booking limits are most effective when some classes can be shut down completely while the rest can be controlled independently of each other. However, in many cases the performance of QAS deteriorates with an increase in the arrival rate while QA improves or remains stable. This indicates that a nonseparable value function approximation has some advantages in the overall performance and robustness of resulting control policies.

The sources of improvements of QAS and QA over FCFS can be determined by separate examination of average penalties and gross profits per cycle resulting from the control policies. The average cycle penalties incurred

Table 2. Average total cycle penalties incurred under the best QA and QAS policies as a percentage of penalties incurred by the FCFS policy for various levels of flexibility/advance booking discount, δ , capacity uncertainty, σ , and magnitudes of arrival rate, $\bar{\lambda}$.

δ	σ	$\bar{\lambda}$	$i_0 = 4$ (%)		$i_0 = 8$ (%)	
			QAS	QA	QAS	QA
0.2	2	0.05	51.2	21.3	58.3	79.3
0.2	2	0.10	81.8	54.1	93.8	43.5
0.2	2	0.25	65.4	57.9	81.6	88.1
0.2	2	0.50	31.6	54.9	100.2	87.5
0.2	5	0.05	19.7	16.0	18.7	20.7
0.2	5	0.10	20.5	17.8	15.5	17.2
0.2	5	0.25	14.9	21.0	24.0	18.7
0.2	5	0.50	10.9	12.6	23.4	19.0
0.5	2	0.05	64.2	63.0	75.6	66.6
0.5	2	0.10	89.9	47.9	110.3	58.9
0.5	2	0.25	58.4	66.0	55.5	95.2
0.5	2	0.50	27.4	54.3	160.8	66.5
0.5	5	0.05	16.1	16.6	28.9	30.8
0.5	5	0.10	30.2	20.4	32.5	25.9
0.5	5	0.25	19.0	30.9	60.0	45.3
0.5	5	0.50	24.1	30.9	46.3	44.2
Average			39.1	36.6	61.6	50.5

under the QAS and QA policies as a percentage of penalties incurred under FCFS are given in Table 2. The average gross profits per cycle for QAS and QA are given in Table 3 as the percentage difference with FCFS gross profits. We

Table 3. Percentage difference in the average total gross profits per cycle collected under the best QA and QAS policies with the gross profits for the FCFS policy for various levels of flexibility/advance booking discount, δ , capacity uncertainty, σ , and magnitudes of arrival rate, $\bar{\lambda}$.

δ	σ	$\bar{\lambda}$	$i_0 = 4$ (%)		$i_0 = 8$ (%)	
			QAS	QA	QAS	QA
0.2	2	0.05	5.9	4.5	6.9	12.8
0.2	2	0.10	17.5	16.0	10.2	10.7
0.2	2	0.25	11.8	19.0	9.7	15.2
0.2	2	0.50	7.8	20.0	10.1	14.1
0.2	5	0.05	-8.8	-9.3	-14.5	-14.8
0.2	5	0.10	-8.2	-7.3	-16.4	-14.3
0.2	5	0.25	-13.6	-7.9	-11.3	-11.5
0.2	5	0.50	-16.3	-10.0	-10.9	-9.4
0.5	2	0.05	26.4	29.8	36.8	35.5
0.5	2	0.10	56.7	47.0	39.6	36.0
0.5	2	0.25	43.7	47.4	30.2	38.9
0.5	2	0.50	35.6	44.3	44.1	37.1
0.5	5	0.05	6.9	6.7	13.1	17.2
0.5	5	0.10	33.2	24.4	12.2	14.5
0.5	5	0.25	18.1	22.3	30.9	29.2
0.5	5	0.50	14.5	23.7	28.6	29.9
Average			14.5	16.9	13.7	15.1

Table 4. Unused capacity (shown as (unused volume, unused weight)) for each flight for the FCFS and the best of QAS and QA policies in the case of four departures per cycle.

δ	σ	$\bar{\lambda}$	FCFS				Best of QAS and QA			
			1	2	3	4	1	2	3	4
0.2	2	0.05	(7.9, 3.3)	(7.0, 3.2)	(7.5, 3.2)	(7.9, 3.3)	(7.6, 3.7)	(7.4, 4.5)	(7.5, 3.6)	(7.9, 4.8)
0.2	2	0.10	(7.9, 4.1)	(7.6, 4.1)	(8.0, 4.1)	(8.2, 4.1)	(6.7, 2.5)	(6.7, 4.5)	(7.4, 3.8)	(8.0, 4.3)
0.2	2	0.25	(7.7, 4.1)	(7.4, 4.1)	(7.6, 4.1)	(8.2, 4.1)	(6.8, 3.4)	(6.6, 4.3)	(6.5, 2.6)	(7.7, 4.1)
0.2	2	0.50	(7.7, 4.1)	(7.4, 4.1)	(7.4, 4.1)	(8.2, 4.0)	(7.4, 3.3)	(6.6, 4.0)	(6.0, 2.7)	(6.5, 4.2)
0.2	5	0.05	(10.5, 7.2)	(10.1, 7.3)	(10.5, 7.3)	(10.7, 7.3)	(11.6, 9.0)	(11.4,10.3)	(11.7, 9.4)	(12.1,10.1)
0.2	5	0.10	(10.2, 7.2)	(10.0, 7.2)	(10.5, 7.3)	(10.5, 7.3)	(11.5, 9.4)	(12.2,10.0)	(11.9, 9.0)	(11.6, 9.9)
0.2	5	0.25	(10.2, 7.2)	(9.8, 7.2)	(10.1, 7.3)	(10.5, 7.2)	(11.4, 8.7)	(12.0, 9.8)	(11.7, 8.9)	(11.6,10.1)
0.2	5	0.50	(10.2, 7.2)	(9.8, 7.2)	(10.0, 7.3)	(10.5, 7.2)	(11.8, 9.2)	(12.5,10.1)	(10.9, 9.0)	(12.2,10.5)
0.5	2	0.05	(8.5, 4.1)	(7.7, 4.0)	(8.2, 4.1)	(8.5, 4.1)	(7.5, 4.0)	(7.3, 5.0)	(6.2, 3.3)	(7.3, 4.3)
0.5	2	0.10	(7.9, 4.1)	(7.6, 4.1)	(8.0, 4.1)	(8.2, 4.1)	(7.0, 3.5)	(8.0, 5.1)	(6.4, 2.7)	(8.4, 5.2)
0.5	2	0.25	(7.7, 4.1)	(7.4, 4.1)	(7.6, 4.1)	(8.2, 4.1)	(8.2, 4.2)	(7.9, 5.4)	(7.4, 3.3)	(8.8, 5.9)
0.5	2	0.50	(7.7, 4.1)	(7.4, 4.1)	(7.4, 4.1)	(8.2, 4.0)	(6.4, 3.2)	(7.0, 4.1)	(6.2, 2.8)	(8.1, 4.1)
0.5	5	0.05	(10.5, 7.2)	(10.1, 7.3)	(10.5, 7.3)	(10.7, 7.3)	(11.4, 9.8)	(12.9,11.3)	(12.3, 9.4)	(12.8,10.5)
0.5	5	0.10	(10.9, 8.2)	(10.7, 8.1)	(11.1, 8.2)	(11.3, 8.1)	(11.8, 9.4)	(12.6,11.1)	(11.6, 9.3)	(12.2, 9.8)
0.5	5	0.25	(10.9, 8.1)	(10.6, 8.1)	(10.6, 8.1)	(11.2, 8.1)	(11.5, 9.0)	(13.6,11.5)	(12.8,10.0)	(11.7,10.2)
0.5	5	0.50	(11.0, 8.1)	(10.6, 8.1)	(10.5, 8.1)	(11.3, 8.1)	(11.4, 9.0)	(12.4,10.1)	(12.0, 9.1)	(12.1, 9.9)

observe that, on average over all experiments, QA policies incur smaller penalties and generate higher gross profits. Moreover, the highest levels of penalties are observed for QAS policies. The effects of QA and QAS policies vary with experimental settings. The penalties are lower relative to FCFS when the level of capacity uncertainty σ is high, and they are especially low when the flexibility/advance booking discount δ is at a low level. For the latter combination of parameters, the average gross profits are below those of FCFS. This indicates that the performance improvement comes specifically from the reduction in penalties in that case. On the other hand, the gains in QAS and QA gross profits are especially high when σ is low and δ is high. For the same experimental settings, QAS and QA penalties are comparatively higher. This indicates that the overall performance gains of QAS and QA come from higher gross profits rather than a reduction in penalties for low σ and high δ . The observed effects in all these cases are quite natural, because it is easier to improve performance by improving gross profits when the capacity uncertainty is low and the flexibility/advance booking discounts are high, and by reducing penalties when their values are the opposite.

A markedly different functioning of the QAS and QA policies in comparison to FCFS shows up in the utilization of flight volume and weight capacities (see Table 4). The weight capacity utilization of each flight (which is a tighter capacity dimension in these examples) is evenly balanced in the case of FCFS because the FCFS policy accepts shipments of different classes in proportion to their effective arrival rate. In contrast, the QAS/QA policy may accept only a small number of shipments in some classes (such as classes requiring travel between the spokes). This explains underutilization of flights originating at the spokes and a heavier utilization of flights from the hub under the

QA policy. In the cases of low capacity uncertainty σ , the QAS/QA policy utilizes hub-bound flights more than FCFS. When the capacity uncertainty is high, QAS/QA uses all the flights less than FCFS in either capacity dimension.

5. Conclusions

The paper presents a new operational model for cargo capacity control on a network. This model explicitly incorporates the uncertainties in available capacity inherent in the cargo shipping business and flexible routing of shipments from origins to destinations. The model falls within a class of Markov decision processes with decisions made at two time scales: very frequent accept/reject decisions for bookings, and infrequent periodic shipping decisions for accepted shipments. To solve the resulting high-dimensional dynamic program, we propose an approximation method based on a quadratic approximation architecture with cross-product terms whose parameters are found via an approximating linear program. The simulation-based solution method can be parallelized. Our numerical illustrations show that the proposed heuristic is competitive with an approach that uses exact calculation of the policy in acceptance control, and that the use of cross-product terms in the quadratic value-function approximation is advantageous in comparison with a separable quadratic approximation.

6. Electronic Companion

An electronic companion to this paper is available as part of the online version that can be found at <http://or.journal.informs.org/>.

Acknowledgments

The support of the Natural Sciences and Engineering Research Council of Canada (grants 261512-04 and 341412-07) is gratefully acknowledged. This work was made possible by the facilities of the Shared Hierarchical Academic Research Computing Network (SHARCNET: www.sharcnet.ca). The fourth author sends prayers of utmost gratitude to God in Whom he sought inspiration during this work and thanks Ying Liu of Royal Military College, Canada, for helpful discussions of the numerical procedure.

References

- Adelman, D. 2007. Dynamic bid-prices in revenue management. *Oper. Res.* **55**(4) 647–661.
- Amaruchkul, K., W. Cooper, D. Gupta. 2007. Single-leg air-cargo revenue management. *Transportation Sci.* **41**(4) 457–469.
- Becker, B., N. Dill. 2007. Managing the complexity of air cargo revenue management. *J. Revenue and Pricing Management* **6**(3) 175–187.
- Bertsekas, D., J. Tsitsiklis. 1996. *Neuro-Dynamic Programming*. Athena Scientific, Cambridge, MA.
- Bertsimas, D., I. Popescu. 2003. Revenue management in a dynamic network environment. *Transportation Sci.* **37**(3) 257–277.
- Blomeyer, J. 2006. Air cargo revenue management. Ph.D. thesis, Tilburg University, Tilburg, The Netherlands.
- de Farias, D., B. Van Roy. 2003. The linear programming approach to approximate dynamic programming. *Oper. Res.* **51**(6) 850–865.
- de Farias, D., B. Van Roy. 2004. On constraint sampling in the linear programming approach to approximate dynamic programming. *Math. Oper. Res.* **29**(3) 462–478.
- Farias, V., B. Van Roy. 2006. An approximate dynamic programming approach to network revenue management. Working paper, Stanford University, Stanford, CA.
- Godfrey, G., W. Powell. 2002. An adaptive dynamic programming algorithm for dynamic fleet management, I: Single period travel times. *Transportation Sci.* **36**(1) 21–39.
- Hellermann, R. 2006. *Capacity Options for Revenue Management: Theory and Applications in the Air Cargo Industry*. Springer-Verlag, Berlin.
- Huang, K., W. Hsu. 2005. Revenue management for air cargo space with supply uncertainty. *J. Eastern Asia Soc. Transportation Stud.* **5** 570–580.
- IATA. 2009. Fact sheet: Industry statistics. http://www.iata.org/pressroom/facts_figures/fact_sheets/.
- Kasilingam, R. 1996. Air cargo revenue management: Characteristics and complexities. *Eur. J. Oper. Res.* **96**(1) 36–44.

- Luo, S., M. Cakanyildirim, R. Kasilingam. 2009. Two-dimensional cargo overbooking models. *Eur. J. Oper. Res.* **197**(3) 862–883.
- Marbach, P., O. Mihatsch, J. Tsitsiklis. 2000. Call admission control and routing in integrated service networks using neuro-dynamic programming. *IEEE J. Sel. Areas Comm.* **18**(2) 197–208.
- Meissner, J., A. Strauss. 2008. Network revenue management with inventory-sensitive bid prices and customer choice. Working paper, Lancaster University, Lancaster, UK.
- Moussawi, L., M. Cakanyildirim. 2005. Profit maximization in air cargo overbooking. Technical report, The University of Texas at Dallas, Dallas.
- Pak, K., R. Dekker. 2005. Cargo revenue management: Bid-prices for a 0-1 multi knapsack problem. Working paper, Erasmus University Rotterdam, Rotterdam, The Netherlands.
- Popescu, A. 2006. Air cargo revenue and capacity management. Ph.D. thesis, Georgia Institute of Technology, Atlanta.
- Popescu, A., P. Keskinocak, E. Johnson, M. LaDue, R. Kasilingam. 2006. Estimating air-cargo overbooking based on a discrete show-up-rate distribution. *Interfaces* **36**(3) 248–257.
- Powell, W. 2007. *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. John Wiley & Sons, Hoboken, NJ.
- Puterman, M. 1994. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, New York.
- Sandhu, R., D. Klabjan. 2006. Fleeting with passenger and cargo origin-destination booking control. *Transportation Sci.* **40**(4) 517–528.
- Simpson, R. W. 1989. Using network flow techniques to find shadow prices for market and seat inventory control. Technical report, MIT Flight Transportation Laboratory Memorandum M89-1, Massachusetts Institute of Technology, Cambridge, MA.
- Slager, B., L. Kapteijns. 2003. Implementation of cargo revenue management at KLM. *J. Revenue and Pricing Management* **3**(1) 80–90.
- Smith, B., C. Penn. 1988. Analysis of alternative origin-destination control strategies. *Proc. 28th Annual AGIFORS Sympos., New Seabury, MA*.
- Talluri, K., G. van Ryzin. 1998. An analysis of bid-price controls for network revenue management. *Management Sci.* **44**(11) 1577–1593.
- Talluri, K., G. van Ryzin. 2004. *The Theory and Practice of Revenue Management*. Kluwer Academic Publishers, Norwell, MA.
- Topaloglu, H. 2008. A stochastic approximation method to compute bid prices in network revenue management problems. *INFORMS J. Comput.* **20**(4) 596–610.
- van Ryzin, G., G. Vulcano. 2008. Simulation-based optimization of virtual nesting controls for network revenue management. *Oper. Res.* **56**(4) 865–880.
- Xiao, B., W. Yang. 2010. A revenue management model for products with two capacity dimensions. *Eur. J. Oper. Res.* **205**(2) 412–421.