

Invited Tutorial

Airline fleet assignment concepts, models, and algorithms

Hanif D. Sherali *, Ebru K. Bish, Xiaomei Zhu

*Grado Department of Industrial and Systems Engineering (0118), Virginia Polytechnic Institute and State University,
Blacksburg, VA 24061, United States*

Received 5 May 2004; accepted 3 January 2005

Available online 2 April 2005

Abstract

The *fleet assignment problem (FAP)* deals with assigning aircraft types, each having a different capacity, to the scheduled flights, based on equipment capabilities and availabilities, operational costs, and potential revenues. An airline's fleet-ing decision highly impacts its revenues, and thus, constitutes an essential component of its overall scheduling process. However, due to the large number of flights scheduled each day, and the dependency of the FAP on other airline processes, solving the FAP has always been a challenging task for the airlines. In this paper, we present a tutorial on the basic and enhanced models and approaches that have been developed for the FAP, including: (1) integrating the FAP with other airline decision processes such as schedule design, aircraft maintenance routing, and crew scheduling; (2) proposing solution techniques that include additional considerations into the traditional fleet-ing models, such as considering itinerary-based demand forecasts and the recapture effect, as well as investigating the effectiveness of alternative approaches such as randomized search procedures; and (3) studying dynamic fleet-ing mechanisms that update the initial fleet-ing solution as departures approach and more information on demand patterns is gathered, thus providing a more effective way to match the airline's supply with demand. We also discuss future research directions in the fleet assignment arena.

© 2005 Elsevier B.V. All rights reserved.

Keywords: Airlines; Fleet assignment; Integer programming; Large scale optimization; Transportation

1. Introduction

Aircraft seats are an airline's product. Similar to any other product, a larger quantity secures sales, while extra inventory incurs costs. For airlines, providing larger capacities implies higher operating costs. On the other hand, aircraft seats are "perishable", that is, unsold seats at the departure of the flight are wasted. Consequently, the ideal strategy is to provide just the "right number" of seats to passengers at the "right

* Corresponding author. Tel.: +1 540 231 5474; fax: +1 540 231 3322.

E-mail addresses: hanifs@vt.edu (H.D. Sherali), eburu@vt.edu (E.K. Bish), xzhu@vt.edu (X. Zhu).

price”. The first of these two ideals is addressed by the fleet assignment process, which is the subject of the present paper, while the second falls in the purview of yield or revenue management.

The *fleet assignment problem (FAP)* deals with assigning aircraft types, each having a different capacity, to the scheduled flights, based on equipment capabilities and availabilities, operational costs, and potential revenues. An airline’s fleet decision highly impacts their revenues: Assigning a smaller aircraft than needed on a flight will result in *spilled* (i.e., lost) customers due to insufficient capacity; assigning a larger aircraft will result in *spoiled* (i.e., unsold) seats, and presumably higher operational costs. Thus, FAP constitutes an essential component of an airline’s overall scheduling process. However, due to the large number of flights scheduled each day, which can easily reach thousands for a major airline, and the dependency of the FAP on other airline processes such as schedule design, crew scheduling, aircraft routing, maintenance planning, and revenue management, solving the FAP has always been a challenging task for the airlines. As a result, it is not surprising that the FAP has been extensively studied in the Operations Research literature. However, most of the traditional approaches proposed for the FAP depend on solving the FAP in isolation from the other airline scheduling processes and under restrictive assumptions such as considering the same-every-day schedule; and using point forecasts for flight-based demands instead of itinerary-based or path-based demands. Furthermore, customers rejected from their requested itineraries (due to capacity restrictions) are often assumed to be lost. In reality, they may choose to take another route that is comparable to their first choice itinerary in terms of the origin, destination, and time-frame (i.e., they may be *recaptured*). The recapture effect has mostly been ignored in analytical studies until very recently.

Recent advances in information technology, coupled with an increasingly competitive marketplace, have motivated researchers to consider new approaches for the FAP over the last decade. These new directions include: (1) integrating the FAP with other airline decision processes such as schedule design, aircraft maintenance routing, and crew scheduling; (2) proposing solution techniques that include additional considerations into the traditional fleet decision models, such as considering itinerary-based demand forecasts and the recapture effect, as well as investigating the effectiveness of alternative approaches such as randomized search procedures; and (3) studying dynamic fleet decision mechanisms that update the initial fleet decision solution as departures approach and more information on demand patterns is gathered, thus providing a more effective way to match the airline’s supply with demand.

The aim of this paper is to present a tutorial on the basic and enhanced models and approaches that have been developed for the FAP as well as to suggest some future research directions in this arena. For the sake of exposition, we focus on several key recent papers on the FAP, instead of providing an exhaustive survey in this area, for which we refer the interested readers to the review articles by [Gopalan and Talluri \(1998\)](#), [Yu and Yang \(1998\)](#), [Barnhart et al. \(2004\)](#), [Clarke and Smith \(2004\)](#), and [Klabjan \(in press\)](#). A special issue of Transportation Science ([Ball, 2004](#)) also provides a variety of topics on aviation operations research.

The remainder of this paper is organized as follows: We first introduce the terminology that will be used throughout this paper in Section 2. We then proceed to describe the various fleet assignment models (FAM) in the subsequent sections. Specifically, the basic FAM as well as the proposed solution approaches are discussed in Section 3. Section 4 presents various approaches that aim at integrating the FAM with other airline decision processes. Section 5 addresses enhanced FAMs that include certain additional considerations, and Section 6 covers some dynamic fleet decision models. Finally, Section 7 concludes this paper with some recommendations for future research directions in this area.

2. Terminology

Fleet type (aircraft type): A certain model of aircraft, such as Boeing’s B767-300. All aircraft of the same type have the same cockpit configuration, crew qualification requirements, maintenance requirements, and capacity.

Fleet family (aircraft family): A set of aircraft types, each having the same cockpit configuration and crew qualification requirements. Thus, the same crew can fly any aircraft type of the same family. An example of an aircraft family is the Boeing 757/767 family, which consists of multiple aircraft types, such as the B757-200 and the B767-300, having capacity ranges between 186 and 255 passengers.

Leg (flight leg): An airport-to-airport flight segment that starts at a specific departure time and connects two stops of a flight, i.e., a leg spans the journey from the time an aircraft takes off until it lands.

Path (itinerary): A sequence of one or more flight legs between a specific origin and destination, starting at a specific departure time. Thus, there can be multiple paths between each origin–destination pair.

Through-flight: Two or more legs that are desirable to be flown by the same aircraft. Through flights are attractive to customers who fly multiple legs between their origins and destinations, because, even though the aircraft makes intermediate stops, they can stay on the same aircraft until they reach their final destination.

O–D: An origin–destination pair corresponding to a path.

Fare class: A particular type of fare restriction. For example, a *Y* fare is the unrestricted fare (i.e., after purchase, the departure day can be changed with no penalty), whereas a *W* fare is more restricted (i.e., the departure day can be changed only by incurring a penalty, and the ticket should be purchased at least two weeks in advance of flight departure).

Turn-time: The minimum time an aircraft needs between its landing time and the next take-off time. This includes the time for some minor inspections, preparation of the aircraft for its next trip, and its movement on the runway. The turn-time is aircraft- and airport-dependent, and typically equals 30–40 minutes for domestic flights.

3. Basic fleet assignment models

In this section, we describe the basic fleet assignment models that lay the foundation of analytical work in this field. Clearly, higher revenues can be realized by fleetting differently every day of the week since demands may vary over the different days of the week. However, this extra flexibility increases the computational complexity significantly at the fleetting stage as opposed to using the same fleet assignment every day of the week (“the same-every-day fleet assignment”); this will be further discussed in Section 5.2. Consequently, most fleet assignment studies consider the same-every-day fleetting decisions, and all the models discussed in this paper are based on this concept, except for the ones proposed by Barnhart et al. (1998) (Section 4.2) and by Bélanger et al. (2004a,b) (Section 5.2).

The fleet assignment problem is typically formulated as a mixed-integer program, based on an airline’s flight network. There are two principal trends that are adopted in constructing the network: using the arcs to represent connections (*connection networks*), and using the arcs to represent flight legs (*time–space networks*). In essence, these two constructs are similar in that they both assure that the model abides by the following main constraints: (1) *cover constraints* so that each leg is assigned to exactly one fleet type; (2) *balance constraints* for conservation of flow; and (3) *aircraft availability constraints* whereby the number of available aircraft of each type bounds their usage. However, because of the differences in the interpretation of arcs in these two networks, the constraints in the resulting mathematical formulations are slightly different. Also, for this same reason, their mixed-integer formulations enable different types of branching strategies.

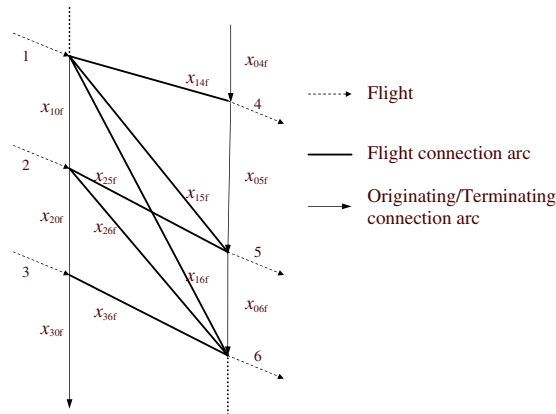
In the following subsections, we review two classical formulations based on their underlying network representations.

3.1. Basic FAM using a connection network structure

Abara (1989) was one of the first researchers to address realistically sized fleet assignment problems using a *connection*-based network structure. In this network, the nodes represent the points of time when

flights arrive or depart. In addition, an imaginary master source node and a master sink node are conceptualized (not actually created) in the network to account for the beginning-of-the-day and the end-of-the-day effects. There are three types of arcs representing the different types of connections: the *flight connection arcs* link the arrival nodes to the departure nodes, the *terminating (connection) arcs* link the arrival nodes to the master sink node to represent aircraft arriving and remaining at the station for the rest of the day, and the *originating (connection) arcs* link the master source node to the departure nodes to represent the aircraft that are present at the station at the beginning of the day. All flight connections have to be *feasible* with respect to flight arrival and departure times; that is, the minimum turn-time has to be observed between the arrival flight and the following departure flight to allow for the connection. The binary decision variables correspond to fleet types that cover these three types of connections at stations. Fig. 1, taken from Abara (1989), illustrates the connection network for a station that includes 12 connections: six (feasible) flight connections, three connections from arrival flights to terminations, and three connections from originations to departure flights.

Next, we introduce some notation to present Abara's mathematical formulation. Let L be the set of flight legs, indexed by i and j , F be the set of fleet types, indexed by f , and S be the set of stations, indexed by s . Further, let A_s and D_s be the respective sets of arrival and departure legs for station s , $s \in S$. Define x_{ijf} to be a binary decision variable that takes on a value of 1 if the feasible connection between leg $i \in L$ to leg $j \in L$ is covered by fleet type $f \in F$, and is 0 otherwise. (The indices $i = 0$ and $j = 0$ denote originating and terminating arcs, respectively.) The three imperative constraints (cover, balance, availability) apply. The objective is to maximize the expected revenue less the operating cost. The benefit from using fleet type f for a connection is arbitrarily assigned as the benefit from the departing flight j of the connection, and is denoted as p_{jf} , which is a comprehensive combination of profit, aircraft utilization, etc. Abara (1989) uses a nominal unit operating cost, denoted by c , for each assignment of the type x_{0jf} that initiates the use of a fleet type f for some flight leg j . In general, however, this cost factor c can be made fleet and flight leg specific, and could be computed by combining the operating costs with the cost from spilled passengers that are estimated from the projected demands, recapture rates, fare structure, and available seats. The practice of calculating the cost is not universal within the industry. We will discuss this issue further in Section 5.1.



Arrow arcs 1–6 represent flights. They are not included in the decision variables. The decision variables are the binary “connection” variables x_{ijf} for fleet type f to cover connection i to j , where x_{i0f} represent the terminating connections and x_{0jf} represent the originating connections.

Fig. 1. Twelve feasible connections in one station from Abara (1989).

The mathematical program based on this connection network is given as follows:

Model 3.1. *Basic fleet assignment model using a connection network*

$$\text{Maximize} \quad \sum_{i \in L \cup \{0\}} \sum_{j \in L} \sum_{f \in F} p_{jff} x_{ijf} - c \sum_{j \in L} \sum_{f \in F} x_{0jf} \quad (1a)$$

$$\text{subject to} \quad \sum_{i \in L \cup \{0\}} \sum_{f \in F} x_{ijf} = 1 \quad \forall j \in L, \quad (1b)$$

$$\sum_{i \in L \cup \{0\}} x_{ilf} - \sum_{j \in L \cup \{0\}} x_{ljf} = 0 \quad \forall l \in L, f \in F, \quad (1c)$$

$$\sum_{i \in D_s} x_{0if} - \sum_{i \in A_s} x_{i0f} = 0 \quad \forall s \in S, f \in F, \quad (1d)$$

$$\sum_{i \in L} x_{0if} \leq A_f \quad \forall f \in F, \quad (1e)$$

$$x \text{ binary.} \quad (1f)$$

In this model, the cover constraint (1b) requires that each flight is preceded by an arrival or an originating arc that is covered by a fleet type. The equality constraint can be relaxed as $\sum_{i \in L \cup \{0\}} \sum_{f \in F} x_{ijf} \leq 1 \quad \forall j \in L$, if not all flights are required to be served, as in Abara (1989). The balance constraint (1c) assures the flow balance at each leg in the network for each fleet type. In addition, the *schedule balance* constraint (1d) ensures that the same number of aircraft of each type remain at each station every night so that the same assignment can repeat daily. The availability constraint (1e) limits the number of aircraft used to A_f , the number of aircraft available for type f . Other side-constraints such as a limit on the number of aircraft that stay overnight at each station can be added as needed.

In Abara's solution approach, the constraints (1d) and (1e) are chosen to be "soft" constraints, and are relaxed with appropriate penalty functions added to the objective function. That is, (1d) induces a penalty term pertaining to the shortage of originating and terminating fleet types at each station, while (1e) induces a penalty term pertaining to the number of extra aircraft used over the available aircraft. The objective thus maximizes the expected revenue minus the operating cost along with the penalties from the relaxed constraints.

As mentioned above, all feasible connections have to be specified in the network for this model. This ensures that all connections selected in a solution are feasible. As a result, however, the network can easily grow to an unmanageable size due to the large number of possible flight connections. Abara (1989) deals with this problem by specifying a limit on the number of connection variables that are considered for each flight.

Using a very similar model formulation, Rushmeier and Kontogiorgis (1997) employ some preprocessing techniques to solve the problem more effectively without having to specify feasible connections for each flight. In particular, they resort to *connecting complexes* for representing feasible connections whereby the operations at each station are partitioned into complexes such that each complex has an equal number of incoming and outgoing legs, and feasible connections exist only within the same complex. Rushmeier and Kontogiorgis (1997) also consider some additional crew-based side-constraints in their model. They design a heuristic to solve the problem in which the LP (linear programming) relaxation is first solved and the resulting solution is rounded to obtain an initial solution that is fed into a depth-first branch-and-bound process, which is run for a two-cpu-hour time limit. Seven problems having eight aircraft types, over 1600 legs, and nearly 100 stations were solved using this approach in concert with CPLEX 3.0 on an IBM RS-6000/590 workstation having 256 MB of RAM. Within the 2-hour limit, an average of 13 integer feasible solutions per run were discovered for these test problems.

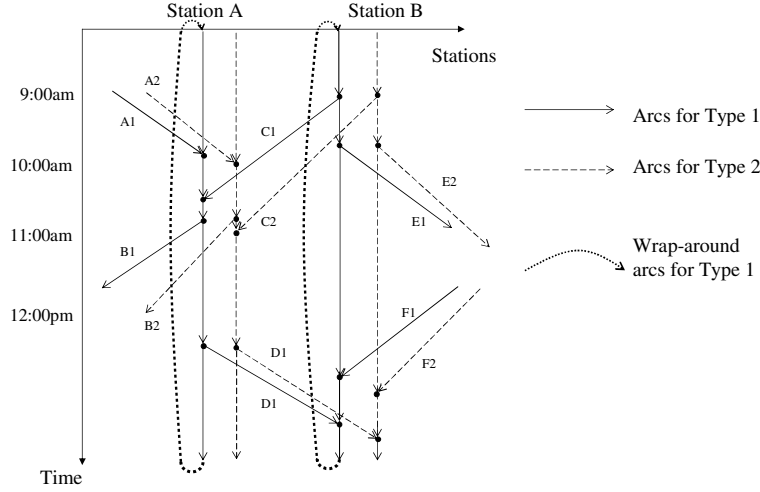
3.2. Basic FAM using a time–space network structure

In contrast with the “connection network”, the *time–space network* structure focuses on representing flight legs, and leaves it to the model to decide on the connections, as long as these are feasible to the time and space considerations. This provides a greater freedom for establishing connections, while reducing the number of decision variables because the number of flight legs is far lesser than the number of possible connections. As Rushmeier and Kontogiorgis (1997) point out, however, the time–space network is not able to distinguish among the specific aircraft on the ground, which limits its application in the subsequent routing problem. Moreover, the use of connection networks can facilitate a different class of branching decisions based on whether or not to select a particular fleet type to make a connection between a specific pair of flight legs. Nonetheless, beginning with Berge and Hopperstad (1993) and Hane et al. (1995), who were among the first researchers to use this network structure, this time–space representation has largely become the method of choice in formulating the fleet assignment problem.

The time–space network representation superimposes a set of networks, one for each fleet type. This allows for fleet type-dependent flight-times and turn-times. If the flight- and turn-times are not significantly different among some particular fleet types, then a composite network representation can be constructed for these types. In each type’s network, each event of flight arrival or departure at a specific time is associated with a node. In order to allow for feasible aircraft connections, an arrival node is placed at the flight’s ready-time, given by its arrival time plus the necessary turn-time, thus representing the time the aircraft is ready to take-off. There are three types of arcs in the network for each fleet type: *ground arcs* representing aircraft staying at the same station for a given period of time, *flight arcs* representing flight legs, and *wrap-around arcs* (or *overnight arcs*) connecting the last events of the day with the first events of the day, which, due to the *same daily schedule*, replicate the first events of the following day. This “wrap-around” ensures continuity of the fleet assignment every day. Airport stations are also duplicated into multiple copies, one for each fleet type. In the sub-network for any given fleet type, a *network time-line* is associated with each station, consisting of a series of event nodes that occur sequentially with respect to time at this station, along with the ground and wrap-around arcs that link these event nodes. The network time-lines of the same fleet type are connected by incoming and outgoing flight arcs to represent the flight-leg-based flows for that type. The requirement that only one fleet type is assigned to each flight leg jointly regulates the flows that occur in the networks of the different types, thus making all the networks interdependent.

Fig. 2 depicts two stations in a time–space network for two fleet types. The time axis progresses downward through the figure, and each node representing a flight arrival or departure event at a particular station is positioned vertically according to its time of occurrence. There are two stations A and B depicted in this figure and two fleet types (Type 1 and Type 2). The full arrows as defined in the figure pertain to the fleet of Type 1 and the broken or dashed arrows pertain to the fleet of Type 2. (Other details of this figure are explained in its legend.)

In the model formulation, the flow on any flight arc is represented by a binary decision variable that is restricted so that the corresponding flight leg is covered by only one type of aircraft. The flows on the ground arcs and the wrap-around arcs take integer values, which represent the number of aircraft of the corresponding type that continue to reside at the particular station. As usual, the formulation includes the three main constraints: cover, balance, and availability. A particular *count time-line* is also specified in the network for counting aircraft, typically at an early time in the morning, say, 3 am EST, when the number of flight arcs is low. For each fleet type, the flows on all the corresponding flight and ground arcs that cross this time-line are summed to assure that the total number of aircraft of this type in use (at this time) does not exceed the number of available aircraft. The network flow balance constraints then ensure that the availability constraints are satisfied for all times in the network. We note here that because of the explicit wrap-around arcs and flow balance restrictions, the schedule balance constraints of Model 3.1 (see



The slanted arrows are the *flight arcs*, the vertical arrows are the *ground arcs*, and the curved arrows are the *wrap-around arcs*. (For clarity of the graph, the wrap-around arcs are shown only for Type 1 and are suppressed for Type 2.) The arc pairs (A_1, A_2) to (F_1, F_2) represent Flights A – F flown by Types 1 and 2, respectively. Here, aircraft of Type 2 require relatively longer turn-times; hence, the arrival times of Type 2 are later than those of Type 1.

Fig. 2. A two-type, two-station fleet-flow time-space network.

(1d)) can be omitted from the formulation for the time-space network. The objective is to maximize the revenue or, equivalently, to minimize the assignment cost.

To be consistent with the existing literature as much as possible, we use the following notation throughout the remainder of this paper.

Notation

S	set of stations in the network, indexed by s , o , or d
F	set of fleet types, indexed by f
L	set of flight legs scheduled, indexed by l or $\{odt\}$, where $o, d \in S$ and t denotes the time when the flight takes off from o or is ready at d for the next take-off
N	set of nodes in the network, indexed by $\{fst\}$, where $f \in F$, $s \in S$, and t denotes the event time
$O(f)$	set of arcs for fleet type f that cross the aircraft count time-line, $f \in F$
c_{fl}	cost of assigning fleet type f to leg l , $f \in F$, $l \in L$
A_f	number of available aircraft for fleet type f , $f \in F$
$x_{fl} =$	$\begin{cases} 1, & \text{if fleet type } f \text{ covers leg } l, f \in F, l \in L \\ 0, & \text{otherwise} \end{cases}$

(The decision variables x_{fl} can also be denoted by x_{fodt} for $f \in F$, $\{odt\} \in L$.)

$y_{fstt'}$	flow of aircraft on the ground arc from node $\{fst\} \in N$ to node $\{fst'\} \in N$ at station $s \in S$ in fleet type f 's network, for $f \in F$, where $t' > t$ in general, and $t' \leq t$ for wrap-around arcs
t^- , t^+	the time preceding and succeeding t , respectively, in the time-line

The aircraft count time-line is used as a starting point for representing the series of events occurring in the network. The first set of nodes after this time-line are denoted as $\{fst_1\}$, $f \in F$, $s \in S$, whereas the last set of nodes of the day are denoted as $\{fst_n\}$, $f \in F$, $s \in S$. With this notation, we are ready to present the basic fleet assignment model, proposed first by Hane et al. (1995):

Model 3.2. *Basic fleet assignment model based on time-space network structure*

$$\text{Minimize} \quad \sum_{l \in L} \sum_{f \in F} c_{fl} x_{fl} \quad (2a)$$

subject to

$$\text{Cover:} \quad \sum_{f \in F} x_{fl} = 1 \quad \forall l \in L, \quad (2b)$$

$$\text{Balance:} \quad \sum_{o \in S} x_{fost} + y_{fst^-t} - \sum_{d \in S} x_{fsdt} - y_{fst^+t} = 0 \quad \forall \{fst\} \in N, \quad (2c)$$

$$\text{Availability:} \quad \sum_{l \in O(f)} x_{fl} + \sum_{s \in S} y_{fst_{nt1}} \leq A_f \quad \forall f \in F, \quad (2d)$$

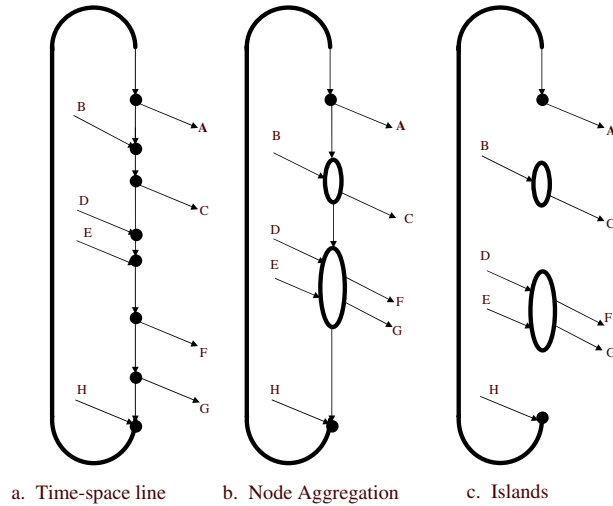
$$x \text{ binary}, \quad y \geq 0. \quad (2e)$$

As discussed above, constraints (2b), (2c), and (2d) are the cover, balance, and aircraft availability constraints, respectively. Not surprisingly, solving FAP on a network consisting of hundreds of stations interwoven by thousands of flights poses a challenging task. Indeed, Gu et al. (1994) have shown that this problem, even without the availability constraints, is NP-hard for three fleet types. Therefore, Hane et al. (1995) have proposed a series of preprocessing steps that aim to reduce the size of the network, and hence the computational effort. In the following, we review these preprocessing steps, which have now become a standard practice.

The first preprocessing step is based on the observation that as long as the network represents the correct connections, the exact time pertaining to each node's event does not matter. Hence, consecutive arrivals and the subsequent consecutive departures can share a single node such that each arrival at the aggregated node can be feasibly connected to any departure at this node. This is called *node aggregation* (a similar technique was adopted earlier by Berge and Hopperstad (1993)). Fig. 3 illustrates this concept. Part (a) of Fig. 3 delineates a network time-line representation of a station for one type. Again, time increases downward in the vertical direction for representing the time of occurrence of each event (node). Part (b) demonstrates the node aggregation that can be performed for this station. Note that there are no more aggregation opportunities available at this station. For example, arrival arc B cannot share the same node as departure arc A, since A departs earlier than B arrives, and thus a connection from B to A would be infeasible. The experiments by Hane et al. (1995) show that node aggregation reduces the number of rows by a factor of 3–6 and the number of columns by a factor of 1–3 in their model.

The second preprocessing step is based on the observation that some stations, especially the spoke stations in hub-and-spoke configurations, have sparse flight activities, leaving no aircraft on the ground during certain periods of time. In this case, the ground arcs have zero flows and can therefore be deleted from the network. For each ground arc removed, there is another zero-valued ground arc, which is encountered after an equal number of arrivals and departures. This ground arc can also be removed. This simplification results in the creation of *islands* in the time-line of the station. (This concept is similar to that of *connection complexes* discussed in Section 3.1.) In Fig. 3, removing the zero-flow ground arcs from part (b) results in the islands shown in part (c).

The third preprocessing technique *eliminates missed connections*. If two flights that must be flown consecutively result in a missed connection when assigned to a fleet type, usually because of the longer turn-times for that fleet type, this pair of flights can be removed from that type's network. In Fig. 2, for example,



- a. The original network time-line representing a station having eight flights A–H.
 b. The same network time-line after node aggregation, where the arrival node for B and the departure node for C are aggregated into one node, and similarly, the nodes for D, E, F, and G are aggregated into a single node.
 c. Deleting the 0-valued ground arcs results in two islands in the network time-line.

Fig. 3. Network reduction for a time-space network.

if flight arcs $C2$ and $B2$ must be flown consecutively, they would result in a missed connection, and so, these two arcs can be removed from Type 2's network.

These three preprocessing steps can considerably reduce the problem size. Indeed, Hane et al. (1995) report that the size of a typical problem instance is reduced from 48,982 rows and 66,942 columns to 7703 rows and 20,464 columns through these three steps. In addition, a combination of algorithmic strategies such as interior-point methods, dual steepest-edge simplex approach, and branching on cover constraints by prioritizing variables based on the objective coefficients' ranges, have been shown to significantly reduce the overall computational requirements.

These basic fleet assignment models have had a major impact on the airline industry. Abara's (1989) implementations have resulted in a 1.4% improvement in operational cost margins at American Airlines. Rushmeier and Kontogiorgis (1997) report an annual benefit of at least \$15 million at US Airways, and the network processing techniques by Hane et al. (1995) have been widely applied in the industry. These early achievements have inspired researchers to aim for further improvements to the FAM.

4. Integrated fleet assignment models

Although the earlier literature reviewed in the previous section attempts to solve the FAP independently of the other airline decision processes, the FAP is, in fact, intricately interwoven with several related overall planning and operational contexts. The FAP depends on the *flight scheduling* process (also called *scheduling/network design*) that specifies the flight network, including the departure and arrival stations and times. The fleet type assignment information produced by the FAP is, in turn, fed into the aircraft *maintenance routing* (or *rotation*) process so that individual aircraft can be assigned specific routes among those prescribed for

its own type, while satisfying maintenance constraints. The *crew scheduling* process is also dependent on the fleet assignment solution in that each crew member needs to be assigned to legs that are flown by the fleet types that he/she is qualified to fly. Furthermore, the revenue management and the fleet decisions are also interdependent, since revenue management provides the FAP with the estimated revenue or profit parameters in the objective function, while the capacities assigned by FAP restrict the actual revenue realization. Each of these problems, by itself, constitutes an interesting topic for research. We refer the interested reader to several survey papers for more information on these problems; see, for instance, [Gopalan and Talluri \(1998\)](#) for the general airline scheduling process; [Barnhart et al. \(2003\)](#) for the crew scheduling problem; and [Yu and Yang \(1998\)](#) and [Barnhart et al. \(2004\)](#) for applications in revenue management, scheduling processes, along with other studies in the airline industry, such as traffic control.

Clearly, a main drawback of solving these problems sequentially, and thus separately, is that an optimal solution for one problem is not necessarily optimal for the entire system, and can even yield an infeasible input to the subsequent processes. The interdependence between these processes have motivated researchers to focus on integrated models that simultaneously consider several of these problems so as to achieve a better solution for the entire system. Although no attempts have been made to integrate all the above-mentioned planning processes due to the potential intractability of the resulting model, various integrated models of two or more sub-problems have been proposed, together with solution techniques that have generated optimal or near-optimal solutions and higher revenues for the airlines. In this section, we review such integrated FAM models: Section 4.1 describes the FAM integrated with schedule design, while Section 4.2 examines the FAM integrated with maintenance routing and crew scheduling.

4.1. FAM integrated with schedule design

Schedule design is usually the first step in the airline operations planning process. It determines the flight network, including flight departure times and departure/arrival stations. The basic fleet assignment models discussed above are then solved based on this network with fixed flight departure times.

Clearly, an integrated fleet assignment and flight scheduling model can increase revenues by allowing for improved flight connection opportunities in the FAP. This is the stance adopted by [Desaulniers et al. \(1997\)](#) and [Rexing et al. \(2000\)](#), who assume that the flight origin–destination information is given, but they allow the flight departure times to vary within certain time-windows, thus resulting in different connection possibilities for the FAP. Denoting the departure time-window for leg i , $i \in L$, as $[a_{if}, b_{if}]$, the necessary turn-time from leg i to leg j , $i \in L, j \in L$, as d_{ijf} , and letting the variable T_{if} be the realized departure time of flight leg i flown by fleet type f , $i \in L, f \in F$, [Desaulniers et al. \(1997\)](#) add the following two constraints to the model of [Abara \(1989\)](#) using a connection network:

$$a_{if} \leq T_{if} \leq b_{if} \quad \forall f \in F \quad \forall i \in L, \quad (3a)$$

$$x_{ijf}(T_{if} + d_{ijf} - T_{jf}) \leq 0 \quad \forall f \in F \quad \forall i, j \in L. \quad (3b)$$

However, the inclusion of the nonlinear constraints (3b) lends computational difficulty and inhibits the model's practical use. Alternatively, using a suitable large number M , (3b) can be reformulated as

$$T_{if} + d_{ijf} - M(1 - x_{ijf}) \leq T_{jf} \quad \forall f \in F \quad \forall i, j \in L, \quad (3c)$$

as used variously in vehicle routing models ([Desrosiers et al., 1983](#); [Kohl and Madsen, 1997](#)).

In the model proposed by [Rexing et al. \(2000\)](#), similar to [Levin \(1971\)](#) (who has considered a single-fleet-type model), the set of departure times for each leg is discretized within its specified time-window and each possible departure time is represented by a copy of the flight arc, so as to accommodate the different departure time possibilities in the time–space network. For the purpose of implementation, they use time-windows of durations 20 and 40 minutes, in which copies of legs are created for every 5- and 1-minute

intervals, respectively. Obviously, only one of these copies needs to be covered by a fleet type. The resulting changes to the basic model formulation presented in Section 3.2 are straightforward: Each variable x_{fl} is replaced with $\sum_{n \in N_{fl}} x_{nfl}$, where N_{fl} denotes the set of arc copies of leg l in fleet type f 's network, and the binary variable x_{nfl} takes the value of 1, if copy n of leg l is covered by fleet type f , and equals 0 otherwise.

Although this representation allows for fleet flexibility, the inclusion of several copies of each arc in the formulation naturally increases the number of decision variables, and hence, the problem size. Two solution approaches are proposed by Rexing et al. (2000): a direct solution technique (DST), which simply uses a commercial package to solve the preprocessed problem, and an iterative solution technique (IST), which we describe in detail below. In both approaches, the network is preprocessed by applying node aggregation and island formation techniques as described in Section 3.2. In addition, *redundant arcs* that stem from the existence of multiple copies of arcs are identified and deleted, whereby all arcs pertaining to the same leg, having the same cost, and sharing the same tail node are dominated by the earliest such one in the time-line. Thus the remaining copies of such arcs can be removed from the network. This pre-processing step of eliminating redundant arcs reduces the problem size by respectively 40% and 66% for sample problems having 5- and 1-minute time intervals within respective time-windows of 20 and 40 minutes. The node aggregation step is shown to further reduce over 80% of the rows and about 56% of the columns. Furthermore, problems using islands are solved 10–20% faster than those without islands.

For the purpose of solving relatively larger practical sized problems, an iterative solution technique (IST) is proposed that is more robust and has lesser memory requirements. In this approach, a master problem and a group of sub-problems are solved iteratively, where the master problem searches for “super-optimal” solutions and the sub-problems check for feasibility. The master problem is formulated by replacing the set of arc copies belonging to each leg with a single arc having a *reduced duration*, which departs at the end of the departure time-window and arrives at the start of the arrival time-window. Because of the existence of the reduced-duration arcs, the optimal solution to the master problem provides a lower bound to the original problem. The feasibility of this solution is then checked by formulating a sub-problem for each fleet type using the original network with time-windows, including only copies of flight arcs that are assigned to this fleet type in the master problem, and adding backward connection arcs that link the later arrivals with the earlier departures at the same station (see Fig. 4). Then, zero costs are associated with the flight arcs and positive penalty costs with the backward arcs, and the new network is solved to minimize the penalty costs on the backward arcs. If the solution yields a zero objective value, then the master problem is feasible for this fleet type. Otherwise, some flight “back-ups” are necessary. The flights that are connected by the backward arcs are identified as the “problem flights”, of which the reduced-duration arcs are replaced by their original copies within the corresponding time-windows. This updated network is passed back to the master problem for the next iteration. The master problem solution yields an optimal assignment when no problem flights arcs are identified in the sub-problems.

Numerical experiments conducted by Rexing et al. (2000) suggest that the IST algorithm performs better than the DST when the time-windows are narrow. However, it is not a good choice when the time-windows are wide (e.g. 40 minutes), since the reduced-duration arcs from the master problem yield many infeasible solutions, and so, several more iterations are typically required. Rexing et al. (2000) report that this extra departure time flexibility added to the FAP results in a cost saving of over \$20 million annually for 20-minute time-windows, and even more for 40-minute time-windows. A further integration of scheduling and fleet flexibility certainly deserves more investigation.

If a tentative flight schedule is at hand that includes certain optional legs, then the decision on which optional legs to offer can be made concurrently with the FAM. Changes in the schedule naturally affect demands. For example, the deletion of a leg may increase the demands on paths having origins, destinations, and time-frames that are compatible with the original paths that contain the particular leg. Therefore, in order to determine the set of optional legs to offer, we need to consider the revenue changes due to the deletion of paths (caused by the deletion of optional legs on these paths). This idea was proposed by Lohatepanont and Barnhart (2004), in which leg selection decisions and FAM are integrated.

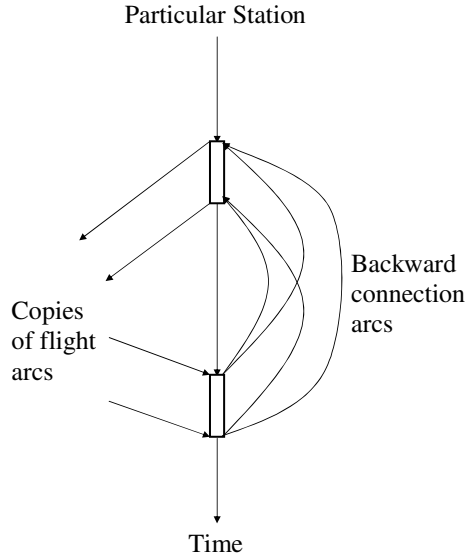


Fig. 4. Backward connection arcs in the sub-problems.

In addition to the notation used in Section 3.2, the following notation is needed to incorporate the leg selection decision into the FAP, as proposed by Lohatepanont and Barnhart (2004).

Π	set of all paths, indexed by i
Π^O	set of paths containing the optional legs (hence, subject to deletion), indexed by i
L^F	set of mandatory legs, indexed by l
L^O	set of optional legs that are candidates for deletion, indexed by l
$L(i)$	set of legs on path i , $i \in \Pi$
r_i	incremental revenue loss if path i is excluded from the flight network, $i \in \Pi^O$
$z_i =$	$\begin{cases} 1, & \text{if path } i \text{ is included in the flight network, } i \in \Pi^O \\ 0, & \text{otherwise} \end{cases}$

Model 4.1. *Integrated leg selection and fleet assignment model*

$$\text{Minimize} \quad \sum_{l \in L} \sum_{f \in F} c_{fl} x_{fl} + \sum_{i \in \Pi^O} r_i (1 - z_i) \quad (4a)$$

$$\text{subject to} \quad \text{balance (2c), availability (2d), plus:} \quad (4b)$$

$$\sum_{f \in F} x_{fl} = 1 \quad \forall l \in L^F, \quad (4c)$$

$$\sum_{f \in F} x_{fl} \leq 1 \quad \forall l \in L^O, \quad (4d)$$

$$z_i - \sum_{f \in F} x_{fl} \leq 0 \quad \forall i \in \Pi^O, l \in L(i), \quad (4e)$$

$$z_i - \sum_{i \in L(i)} \sum_{f \in F} x_{fl} \geq 1 - |L(i)| \quad \forall i \in \Pi^O, \quad (4f)$$

$$(x, z) \text{ binary, } y \geq 0. \quad (4g)$$

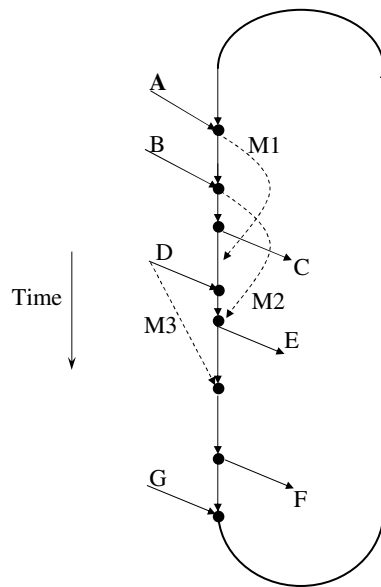
Notice that the cover constraints (2b) are split into (4c) and (4d) to distinguish between the mandatory and optional leg sets. Constraint (4e) ensures that if any leg is excluded from the network, then the path that contains the leg will be excluded, and (4f) ensures that if all the legs contained in a path are included in the network, then the path that contains them is also included. The work by Lohatepanont and Barnhart (2004) also describes detailed calculations for the revenue changes associated with the deletion of a leg. We will discuss this feature in Section 5.1 where we address other related cost considerations.

We note here that if $L^F = \emptyset$, then this model would serve the purpose of designing the entire flight schedule, which is also the idea in Yan and Tseng (2002), whose model integrates the scheduling process with fleet assignment, while including path-based demand considerations. Details of this model are also included in Section 5.1.

4.2. FAM integrated with maintenance, routing, and crew considerations

To assure safety, aircraft need to undergo regular maintenance checks. FAA requires four types of maintenance checks, labeled A, B, C, and D. Among them, the C and D types of checks take longer than 24 hours. FAM deals with these two types of checks by simply reducing the number of available aircraft during the times when some aircraft are scheduled for these maintenance checks. For the other two types of checks, the A checks take around 4 hours, and the B checks take 10–15 hours to perform. These checks can therefore be included in a more realistic, expanded formulation of the FAM.

Clarke et al. (1996) classify the A and B checks as the *short maintenance* and the *long maintenance* checks, respectively, and include these maintenance constraints in Model 3.2. For this purpose, they add two types of maintenance arcs, one for each maintenance check, in the time–space network. Aircraft that are scheduled to undergo maintenance checks in-between their flight tasks should be assigned to these maintenance arcs (see Fig. 5 for an example). Two corresponding lists are also constructed for these scheduled



Arcs $M1$ and $M2$ are *leapfrog arcs* to accommodate long maintenances, and flight arcs D and $M3$ are split arcs, where $M3$ includes additional short maintenance time.

Fig. 5. A time-line with maintenance arcs from Clarke et al. (1996).

maintenance activities. They include, for each maintenance activity, the number of aircraft required, the maintenance station, the aircraft type, the maintenance time-window (i.e., the earliest starting time and the latest completion time for that maintenance activity), and its expected duration. If a flight arrives at a maintenance station after the earliest starting time of the corresponding time-window and a long maintenance can be completed within the time-window, then a *leapfrog arc* is created, which departs from the arrival event node and arrives at the same station after the corresponding maintenance duration (see arcs M1 and M2 in Fig. 5). Multiple leapfrog arcs can be created for a maintenance activity, depending on the time-window and the duration of the maintenance service. For example, arcs M1 and M2 in Fig. 5 correspond to the same maintenance activity.

Let PL denote the set of long maintenance activities, indexed by p , and for activity $p \in PL$, let M_p be the number of aircraft required to undergo long maintenance and $J(p)$ be the set of eligible leapfrog arcs, indexed by j . Furthermore, let m_{pj} be the number of aircraft taking arc $j \in J(p)$. Then, the long maintenance constraints can be stated as

$$\sum_{j \in J(p)} m_{pj} = M_p \quad \forall p \in PL. \quad (5)$$

Note that similar to the ground arcs, the leapfrog arcs also participate in the network flow balance constraints (2c) as well as in the aircraft availability constraints (2d).

The duration of each long maintenance arc is, generally, longer than half of its allocated time-window length. This ensures that no aircraft can take more than one long maintenance arc—corresponding to the same requirement—in a row. (For example, in Fig. 5, if the duration for M1 were shorter than half of its time-window, then it could finish before M2 even starts, which might result in an aircraft being assigned to two successive identical maintenance operations.)

On the other hand, most short maintenance requirements can be met simply by having the aircraft stay overnight at a maintenance station. This leads to the following set of constraints:

$$\sum_{s \in CM(f)} y_{fst_n t_1} \geq NM(f) \quad \forall f \in F, \quad (6)$$

where $CM(f)$ is the set of stations that can perform the required maintenance check for type f , and $NM(f)$ is the average number of type f aircraft that need short maintenance. There are circumstances, however, in which simply counting the number of aircraft is not sufficient to ensure the required maintenance restrictions for each aircraft. Some modifications in the network are thus needed. Unlike long maintenance checks, however, short maintenance checks are not necessarily longer than half of their time-windows. Thus, particular attention should be paid to prevent double-counting. For this purpose, each flight arc that can be followed by a short maintenance check is split into two arcs: one is the real flight arc, and the other is the flight arc *extended* by the duration of the short maintenance requirement (see arcs D and M3 in Fig. 5). The cover constraints (2b) in Model 3.2 are thus transformed to ensure that for each flight leg, either its real flight arc or its extended flight arc is covered by an aircraft type:

$$\sum_{f \in F} x_{fl} + \sum_{f \in F} \sum_{p \in PS} x m_{pfl} = 1 \quad \forall l \in L, \quad (7)$$

where PS denotes the set of short maintenance activities, and where $x m_{pfl}$ is a binary variable that takes the value of 1 if fleet type f flying leg l undergoes maintenance p , and equals 0 otherwise. The variable $x m_{pfl}$ is ascribed a smaller objective coefficient than x_{fl} $\forall f \in F, l \in L$, in order to encourage more maintenance services.

This construct, however, significantly increases the number of integer variables in the problem. Alternatively, leapfrog arcs can be added following the arrival flights at the beginning of their short maintenance time-windows. Constraints (7) then revert to the usual cover restrictions (2b) for the legs associated with

such leapfrog arcs. Because the earliest time that double-counting can occur is after the end of the first leapfrog arc, the extended split flight arcs are applied instead of leapfrog arcs after the end of the first leapfrog arc to avoid the double-counting phenomenon. The maintenance constraints then transform to (in notation analogous to that defined before):

$$\sum_{j \in J(p)} (m_{pj} + xm_{pj}) = M_p \quad \forall p \in PS, \quad (8)$$

where xm_{pj} is a binary variable that takes the value of 1 if arc j , $j \in J(p)$, undergoes maintenance p , and equals 0 otherwise, and where the set $J(p)$ now includes the extended split flight arcs, along with the leapfrog arcs, which are created for maintenance activity $p \in PS$.

Clarke et al. (1996) also integrate some crew scheduling considerations into the FAP by attempting to limit the number of *lonely overnights*. This occurs when a crew, which arrives at a non-base station, has to stay overnight at this station because there is no departing aircraft that it is qualified to fly on the same day after the minimum legal rest time. Such overnight rests at non-base stations are very expensive in terms of crew costs. Hence, Clarke et al. (1996) attempt to reduce the time-away-from-base for crews, using a structure similar to the leapfrog arcs.

Using these constructions and the callable library OSL, Release 2, Clarke et al. (1996) solve their model using an LP-based branch-and-bound algorithm, adopting the dual steepest-edge simplex method, and a special ordered set (SOS) branching mechanism, in which the decision variables are prioritized by the type capacity. They are able to thus solve problems involving up to eleven fleet types and 2500 flight legs within 2–5 hours of cpu time on an IBM RS/6000 model 550.

At Delta Air Lines, Subramanian et al. (1994) have incorporated maintenance issues into Coldstart, the fleeting model of Delta, via the use of maintenance arcs in the time-space network, along with the consideration of lonely overnights. They report a projected savings of \$300 million over the initial three years of applying the Coldstart model. However, no related modelling details are provided in the open literature.

Other integrated routing and fleet assignment models include those proposed by Desaulniers et al. (1997) and Barnhart et al. (1998). Both models use the idea of pre-specifying sequences of flights and assigning these sequences to fleet types, and adopt similar branch-and-price schemes, which utilize column generation to provide improved sequences. The difference between the two models is that the one proposed by Barnhart et al. (1998) explicitly accommodates maintenance considerations using ground arcs, while the model of Desaulniers et al. (1997) includes time-windows in the sub-problems using up to two copies of each turn arc. In the following, we use the string-based model of Barnhart et al. (1998) to illustrate the basic concept in these two models. This string-based MIP model solves the weekly (or any cyclic) fleet assignment and aircraft routing problems simultaneously. The basic idea is to pre-specify sequences of maintenance-feasible legs between maintenance stations that satisfy the flow-balance constraints, and assign these *sequences* (instead of single legs) to fleet types. Each such maintenance-feasible sequence is defined as an *augmented string*, which is a flight sequence (*string*) that is extended at the end of the last leg to include the maintenance time at a maintenance station. Thus, upon completing the assignment, a feasible aircraft maintenance routing is at hand.

At each maintenance station, events of arrivals (i.e., maintenance completions) and departures are sorted in increasing order of time. Let $e_{l,a}^f$ and $e_{l,d}^f$ denote the events of arrival and departure, respectively, of leg l at a maintenance station, for fleet type f , $f \in F$. The superscripts “–” and “+” are used to respectively represent the previous and succeeding events. Additionally, the following notation is used in the string-based FAM:

SG	set of augmented strings, indexed by s
SG_l^-	set of augmented strings ending with leg l , $l \in L$
SG_l^+	set of augmented strings starting with leg l , $l \in L$
G^f	set of ground arcs in fleet type f 's network, $f \in F$

a_{ls}	$\begin{cases} 1, & \text{if leg } l \text{ is in augmented string } s, l \in L, s \in SG \\ 0, & \text{otherwise} \end{cases}$
p_j^f	$\begin{cases} 1, & \text{if ground arc } j \text{ crosses the count time-line, } j \in G^f, f \in F \\ 0, & \text{otherwise} \end{cases}$
r_s^f	number of times that augmented string s , assigned to type f , crosses the count time-line, $s \in SG, f \in F$
x_s^f	$\begin{cases} 1, & \text{if augmented string } s \text{ is flown by fleet type } f \text{ (at cost } c_s^f), f \in F, s \in SG \\ 0, & \text{otherwise} \end{cases}$
y_j^f	number of aircraft of type f on the ground arc $j, j \in G^f, f \in F$. (The decision variable y_j^f can also be denoted by $y_{(e,e')}^f$, where e and e' are, respectively, the starting and ending events of the ground arc $(e, e') \in G^f, f \in F$)

The string-based fleet assignment and routing model can then be stated as follows:

Model 4.2. *String-based fleet assignment and routing model*

$$\text{Minimize} \quad \sum_{s \in SG} \sum_{f \in F} c_s^f x_s^f \quad (9a)$$

$$\text{subject to} \quad \sum_{s \in SG} \sum_{f \in F} a_{ls} x_s^f = 1 \quad \forall l \in L, \quad (9b)$$

$$\sum_{s \in SG_l^+} x_s^f - y_{(e_{l,d}^{-f}, e_{l,d}^f)}^f + y_{(e_{l,d}^f, e_{l,d}^{+f})}^f = 0 \quad \forall l \in L, f \in F, \quad (9c)$$

$$- \sum_{s \in SG_l^-} x_s^f - y_{(e_{l,a}^{-f}, e_{l,a}^f)}^f + y_{(e_{l,a}^f, e_{l,a}^{+f})}^f = 0 \quad \forall l \in L, f \in F, \quad (9d)$$

$$\sum_{s \in SG} r_s^f x_s^f + \sum_{j \in G^f} p_j^f y_j^f \leq A_f \quad \forall f \in F, \quad (9e)$$

$$x \text{ binary, } y \geq 0. \quad (9f)$$

In this model, (9b) and (9e) respectively represent the usual cover and availability constraints. Also, notice that instead of focusing on the nodes to maintain the conservation of flow, flow balance is enforced at the first and the last legs of augmented strings, as shown in constraints (9c) and (9d).

Along with the advantage of solving the assignment and routing problems simultaneously, comes the difficulty associated with the large number of strings, which is exponential in the number of legs. As a result, Barnhart et al. (1998) develop a *branch-and-price* approach, in which an LP relaxation is solved at each node of the branch-and-bound tree using column generation (via the “pricing” sub-problem). In this algorithm, the pricing sub-problem is cast as a *resource-constrained shortest path problem* over a maintenance connection network, resembling that of Abara (1989), in which the nodes represent the flights and the arcs represent the connections between flights at the same station. A maximum allowable flying time or elapsed time between successive maintenance operations is considered as a “resource”, and each node and arc in a path respectively consumes this resource by the amount of flying hours and elapsed hours. For maintenance stations, the connection arcs are extended to include the maintenance time. The shortest path problem is solved for each fleet type f . If the shortest path, which in effect determines the reduced cost of the most enterable column with respect to the current LP basis, has a non-negative value for each f , then an optimal LP solution is at hand; otherwise, new columns are generated from strings yielding negative values, and these are added to the master problem in the column generation process.

Using the node and island preprocessing techniques and implementing this branch-and-price approach, good solutions having an optimality tolerance ranging from 0.25% to 1.0% were achieved in about 5 hours on an IBM RS-6000/370 using CPLEX 3.0, for a 7-day, 1,124-flight schedule covering 40 stations and having nine fleet types, containing a total of 89 aircraft.

Based on this type of a string-based model, [Rosenberger et al. \(2004\)](#) propose a robust FAM that creates *rotation cycles* (i.e., sequence of legs assigned to each aircraft) that are short and that involve as few hub stations as possible, so that flight cancellations or delays will have a smaller chance to cascade into a large number of subsequent stations, particularly hubs. In their network, only the time-lines associated with hub stations are included. A string that starts and ends at the same hub is called a *cancellation cycle*, and a string that starts and ends at different hubs forms an *acyclic string*. A network having more cancellation cycles and less acyclic strings is shown to be more robust. As the number of legs in the set of acyclic strings decreases, a lower bound on the number of cancellation cycles increases. Consequently, two alternative models are formed based on the string-based FAM. Both formulations share the same constraints of those in Model 4.2. The first model imposes an upper bound on the number of legs in acyclic strings while minimizing the total cost, and the second model minimizes the number of legs in acyclic strings while constraining the total cost by an upper bound.

5. Fleet assignment models with additional considerations

In addition to the integration efforts discussed in the previous section, researchers have also started relaxing some of the restrictive assumptions made in the basic FAM described in Section 3. In this section, we present some of these efforts. Specifically, Section 5.1 describes an FAM that incorporates path-based demands into the fleetings. Section 5.2 presents a weekly fleetings model. A neighborhood search approach for an FAM that involves through-flight decisions is introduced in Section 5.3. Finally, Section 5.4 discusses some other non-optimization approaches for fleetings problems.

5.1. FAM including passenger considerations

The FAM formulations that we have introduced thus far minimize the total fleetings and spill costs (or maximize the profit), considering a cost term for each possible leg-fleet type combination in the objective function. These costs are calculated based on the corresponding fleet type capacity and the forecasted leg demand. However, a significant portion of all demands flown by US airlines consists of *multiple-leg passengers* (i.e., passengers whose itineraries are comprised of more than one leg), especially under the hub-and-spoke network structure, in which passengers typically fly into and out of hub airports in between their origins and destinations. Clearly, the demand coming from a multiple-leg passenger will be dependent on the availability of seats on all the involved legs (this is called the “*network effect*”). Thus, (i) leg demands can be highly dependent, and (ii) passengers who demand to fly on a particular leg are *not identical* in terms of the revenue that they will generate and the airline resources that they will consume. Therefore, some recent fleet assignment approaches have included the network effect consideration (see, for example [Jacobs et al., 1999](#)). Furthermore, passengers, who fail to get their requested itinerary, can sometimes be routed to other similar paths (the “*recapture effect*”), making path demands dependent.

Traditionally, an isolated path-based passenger decision model (referred to as the “passenger mix” model in the literature) is solved after the FAM, so as to determine which passengers to accept on each leg, given the capacity assigned to each leg as well as path-based demands. [Glover et al. \(1982\)](#), [Dror et al. \(1988\)](#), [Phillips et al. \(1991\)](#), [Farkas \(1995\)](#), [Kniker \(1998\)](#), and [Talluri and Van Ryzin \(1999\)](#) present various versions of the passenger mix model. Below, we present a passenger mix model proposed by [Barnhart et al. \(2002\)](#), which minimizes the loss of revenue due to spills. Consider the following notation:

Π	set of paths in the flight schedule, indexed by i or j
$\Pi(l)$	set of paths in Π that contain leg l , $l \in L$
μ_i	mean demand on path i , $i \in \Pi$
$fare_i$	estimated fare price on path i , given by the weighted average of the different fare-classes, based on the estimated proportion of each fare-class on that path, $i \in \Pi$
\widetilde{Cap}_l	capacity assigned to leg l , $l \in L$
b_i^j	recapture rate; i.e., the fraction of customers spilled from path i toward path j that are <i>successfully</i> captured on path j , $i \in \Pi$, $j \in \Pi$, $j \neq i$
t_i^j	number of passengers spilled from path i who are redirected to (but not necessarily accepted on) path j , $i \in \Pi$, $j \in \Pi$, $j \neq i$
t_i^-	number of passengers spilled from path i that are not recaptured on any other paths, $i \in \Pi$

Barnhart et al. (2002) calculate the recapture rate, b_i^j , using the *quality of service index (QSI)* (Kniker, 1998), which represents the “attractiveness” of each path with respect to all other paths. Denoting the QSI for path j as q_j and the total QSI for all the paths in the market as Q , b_i^j is calculated as $\frac{q_j}{1-Q+q_j}$ $\forall i \neq j$, and is set to 1 for $i = j$. Their proposed formulation can be represented as follows:

$$\text{Minimize} \quad \sum_{i \in \Pi} [fare_i t_i^- + \sum_{j \in \Pi: j \neq i} (fare_i - b_i^j fare_j) t_i^j] \quad (10a)$$

$$\text{subject to} \quad \widetilde{Cap}_l + \sum_{i \in \Pi(l)} (t_i^- + \sum_{j \in \Pi: j \neq i} t_i^j) - \sum_{i \in \Pi} \sum_{j \in \Pi(l): j \neq i} b_i^j t_i^j \geq \sum_{i \in \Pi(l)} \mu_i \quad \forall l \in L, \quad (10b)$$

$$t_i^- + \sum_{j \in \Pi: j \neq i} t_i^j \leq \mu_i \quad \forall i \in \Pi, \quad (10c)$$

$$t \geq 0. \quad (10d)$$

The total lost revenue because of spill is $\sum_{i \in \Pi} (fare_i t_i^- + \sum_{j \in \Pi: j \neq i} fare_i t_i^j)$, while the regained revenue because of recapture is $\sum_{i \in \Pi} \sum_{j \in \Pi: j \neq i} b_i^j fare_j t_i^j$. The difference between these two terms is the net revenue loss due to spillage, which is to be minimized in the objective function. In the capacity constraints (10b), for each leg l , the term $\sum_{i \in \Pi(l)} (t_i^- + \sum_{j \in \Pi: j \neq i} t_i^j)$ is the number of passengers who have requested but are spilled from the paths that contain leg l , and the term $\sum_{i \in \Pi} \sum_{j \in \Pi(l): j \neq i} b_i^j t_i^j$ is the number of passengers recaptured from other paths to the ones that contain leg l . The difference between these two terms is the net spillage from leg l . Therefore, constraints (10b) require that the capacity assigned to any leg cannot be less than the total demand on paths containing this leg minus the net spillage from this leg. Constraints (10c) are the demand constraints, which ensure that the spillage from path i does not exceed the demand on this path.

Integrating (10) with the FAM yields the path-based fleet assignment model proposed by Barnhart et al. (2002).

Model 5.1. Path-based fleet assignment model

$$\text{Minimize} \quad \sum_{l \in L} \sum_{f \in F} c_{fl} x_{fl} + \sum_{i \in \Pi} \left[fare_i t_i^- + \sum_{j \in \Pi: j \neq i} (fare_i - b_i^j fare_j) t_i^j \right] \quad (11a)$$

$$\text{subject to} \quad \text{cover (2b), balance (2c), availability (2d), plus:} \quad (11b)$$

$$\sum_{f \in F} Cap_f x_{fl} + \sum_{i \in \Pi(l)} \left(t_i^- + \sum_{j \in \Pi: j \neq i} t_i^j \right) - \sum_{i \in \Pi} \sum_{j \in \Pi(l): j \neq i} b_i^j t_i^j \geq \sum_{i \in \Pi(l)} \mu_i \quad \forall l \in L, \quad (11c)$$

$$t_i^- + \sum_{j \in \Pi: j \neq i} t_i^j \leq \mu_i \quad \forall i \in \Pi, \quad (11d)$$

$$x \text{ binary}, \quad (y, t) \geq 0. \quad (11e)$$

Note that the capacity constraints (11c) are the key restrictions that link the two separate decision processes together.

Because of the network effect, the inclusion of such demand management decisions greatly increases the size and complexity of the problem. Barnhart et al. (2002) propose a heuristic solution technique for this formulation that is based on considering a restricted set of columns generated by the LP relaxation. Specifically, this approach first omits constraints (11d) from the model, as they are usually not binding. A coefficient reduction technique is applied to constraints (11c) for which $Cap_f > \sum_{i \in \Pi(l)} \mu_i$. This coefficient reduction scheme is more effective when the recapture effects are not considered. The LP relaxation is then solved based on a column and row generation technique, where the columns are generated for spill variables having negative reduced cost, and rows are generated for violated constraints (11d). The resulting model having the generated columns is then solved via a branch-and-bound procedure for determining an integer solution. Their computational experiments using networks having up to 2044 legs and 76,641 paths suggest that incorporating both the network and recapture effects can significantly impact the revenue. Using path-based demand alone yields an annual revenue increase of over \$30 million, and including the recapture effect yields another \$2–\$115 million (in the order of a total of \$33.7 million to \$153.2 million per year) for a major US carrier.

Recall that in Section 4.1, we presented Model 4.1 that simultaneously performs the fleet assignment and leg selection decision, considering a set of optional legs, where this model utilizes an estimate of the revenue difference resulting from the deletion of a leg. We are now ready to calculate the value of this revenue difference. When path j , $j \in \Pi^O$, is deleted from the schedule (due to the deletion of a leg in the path), the direct revenue loss from path j will be $fare_j \mu_j$. However, some customers may choose to select another path, say $i \in \Pi$, thus increasing the demand on path i . Let $\Delta \mu_j^i$ be a demand correction term that denotes the incremental demand on path i when path j is excluded from the network. Correspondingly, the revenue gained on path i because of this demand increase is $fare_i \Delta \mu_j^i$. Thus, the net revenue loss due to the elimination of path j is given by $fare_j \mu_j - \sum_{i \in \Pi: i \neq j} fare_i \Delta \mu_j^i$. Using this new term in the objective function, Lohatepanont and Barnhart (2004) propose to integrate the leg selection decision of Model 4.1 into Model 5.1 as follows:

Model 5.2. *Integrated leg selection and path-based fleet assignment model*

$$\begin{aligned} \text{Minimize} \quad & \sum_{l \in L} \sum_{f \in F} c_{fl} x_{fl} + \sum_{i \in \Pi} \left[fare_i t_i^- + \sum_{j \in \Pi: j \neq i} (fare_i - b_i^j fare_j) t_i^j \right] \\ & + \sum_{j \in \Pi^O} \left(fare_j \mu_j - \sum_{i \in \Pi: i \neq j} fare_i \Delta \mu_j^i \right) (1 - z_j) \end{aligned} \quad (12a)$$

subject to constraints (4b)–(4f) from Model 4.1, plus: (12b)

$$\begin{aligned} & - \sum_{i \in \Pi(l)} \sum_{j \in \Pi^O} \Delta \mu_j^i (1 - z_j) + \sum_{f \in F} Cap_f x_{fl} + \sum_{i \in \Pi(l)} \left(t_i^- + \sum_{j \in \Pi: j \neq i} t_i^j \right) \\ & - \sum_{i \in \Pi} \sum_{j \in \Pi(l): j \neq i} b_i^j t_i^j \geq \sum_{i \in \Pi(l)} \mu_i \quad \forall l \in L, \end{aligned} \quad (12c)$$

$$- \sum_{j \in \Pi^O} \Delta \mu_j^i (1 - z_j) + t_i^- + \sum_{j \in \Pi: j \neq i} t_i^j \leq \mu_i \quad \forall i \in \Pi, \quad (12d)$$

$$(x, z) \text{ binary}, \quad (y, t) \geq 0. \quad (12e)$$

In constraints (12c) and (12d), the terms $\sum_{i \in \Pi(l)} \sum_{j \in \Pi^O} \Delta \mu_j^i (1 - z_j)$ and $\sum_{j \in \Pi^O} \Delta \mu_j^i (1 - z_j)$ are the incremental demands on leg l and path i , respectively, based on the paths excluded from the network.

Model 5.2 is solved using the same heuristic approach proposed for Model 5.1. One additional difficulty of solving Model 5.2 arises due to the estimation of demand correction terms. The demand parameters and the demand correction terms used in the model are estimated via a schedule evaluation package that takes flight schedules as input. Therefore, evaluating demand requires a large number of runs using the package corresponding to all combinations of schedules. To avoid this, demands and demand correction terms are estimated and revised iteratively using the schedule obtained from Model 5.2 along with the schedule evaluation package. Specifically, the demand for the full schedule including all the mandatory and optional legs obtained from the schedule evaluation package are chosen as the initial set of demand estimates. Then, at each iteration, the heuristic approach used for Model 5.1 is applied to solve Model 5.2, using the current demand estimates. The resulting schedule is input to the schedule evaluation package to obtain a new set of demand estimates. These estimated demands are then used to evaluate the revenue by solving a passenger mix model due to [Kniker \(1998\)](#), similar to the one presented in (10). The resulting objective value of this passenger mix model is a maximum schedule revenue. At the end of each iteration, if the difference between this revenue and the estimated revenue obtained from Model 5.2 falls below a pre-specified threshold, or, alternatively, if the solutions produced by Model 5.2 over two consecutive iterations are close enough, then the process is terminated. Otherwise, the paths having inaccurate revenue estimates obtained from Model 5.2 are identified, and the associated demand correction terms are revised based on the set of demand estimates obtained from the schedule evaluation package earlier in the iteration, and the iterations continue using the updated demand correction terms.

Due to the large size of Model 5.2, solving it even with the heuristic proposed for Model 5.1 takes several days. To be able to obtain a solution in a more realistic time-frame, an approximate model is developed by [Lohatepanont and Barnhart \(2004\)](#), in which all demand correction related terms are dropped, resulting in a Model 5.1 type of formulation except that the cover constraints are split as in (4c) and (4d) to treat the sets of mandatory and optional legs separately. Meanwhile, recapture rates are also adjusted to compensate for the effects of schedule changes over demand. When the demand correction terms are present, the recapture rates purely reflect the percentage of passengers that are accommodated by some alternative paths when the capacities on their desired paths are constrained. Now that the demand correction terms are absent, the recapture rates need to also reflect the re-accommodation of passengers from the deleted paths to the remaining paths. Therefore, the set of adjusted recapture rates will be different than the ones calculated using QSI, and are accordingly adjusted iteratively in the same manner as the demand correction terms are adjusted at the end of each iteration, while the stopping criteria are not met. The original (unadjusted) recapture rates are used as the initial values for the adjusted ones. Two practical-size problems are solved using this approach, and yield an annual revenue increase of \$147.5 million and \$360.6 million, respectively.

[Yan and Tseng \(2002\)](#) consider passenger demands in their integrated scheduling and fleet model from a different perspective. Instead of defining passengers' paths to include specific legs, they consider their origins and destinations (O–D pairs), and leave it to the model to select the legs for composing passengers' routes. They resort to two groups of networks. The first group includes the fleet-flow time–space networks, one constructed for each fleet type, where flight arcs are all optional and a decision needs to be made on which ones to select to include into the schedule. The second group of networks are the passenger-flow time–space networks, one for each O–D pair. These networks contain *delivery arcs* and *hold arcs*, which are copies of the flight arcs and ground arcs, respectively, from the fleet-flow time–space networks. In addition, the passenger-flow time–space networks contain *demand arcs*, each starting from the passenger's destination station and arrival time, and ending at the origin station and departure time. The flow upper bounds on these arcs are the forecasted demands. The reason for the particular orientation of these demand arcs is to conserve passenger flows in the network. We use the following notation to present the model by [Yan and Tseng \(2002\)](#):

OD	set of O–D pairs, and thus, the set of passenger-flow networks, indexed by $\{od\}$ or p
PN^p	set of nodes in the passenger-flow network for O–D pair p , indexed by $\{st\}^p$, where $s \in S$, t denotes the event time, and $p \in OD$
DA	set of demand arcs indexed by a or $\{dot\}$, where $\{od\} \in OD$ and t denotes the event time (recall that each demand arc is from the destination station to the origin station)
$\tilde{\mu}_a$	mean demand on demand arc a , $a \in DA$
\widehat{fare}_a	average fare on all paths that can contribute to the demand on arc a , $a \in DA$
$Darc_a$	flow on demand arc a , $a \in DA$
$Larc_l^p$	flow on delivery arc l in network p , $l \in L$, $p \in OD$
$Harc_{stt'}^p$	flow on the hold arc from node $\{st\}^p$ to node $\{st'\}^p$, where $\{st\}^p, \{st'\}^p \in PN^p$, $p \in OD$

The following formulation is based on Yan and Tseng (2002), and is modified for the purpose of consistency in presentation:

Model 5.3. *Integrated scheduling and FAM with passenger demand considerations*

$$\text{Minimize} \quad \sum_{l \in L} \sum_{f \in F} c_{fl} x_{fl} - \sum_{a \in DA} \widehat{fare}_a Darc_a \quad (13a)$$

$$\text{subject to} \quad \text{balance (2c) and availability (2d), plus :} \quad (13b)$$

$$\sum_{f \in F} x_{fl} \leq 1 \quad \forall l \in L^0, \quad (13c)$$

$$\sum_{\substack{d \in S: \\ \{sd\}=p}} Darc_{dst} + \sum_{\substack{s' \in S: \\ s' \neq s}} Larc_{s'st}^p + Harc_{st-t}^p - \sum_{\substack{o \in S: \\ \{os\}=p}} Darc_{sot} - \sum_{\substack{s' \in S \\ s' \neq s}} Larc_{ss't}^p - Harc_{stt'}^p = 0$$

$$\forall \{st\}^p \in PN^p, p \in OD, \quad (13d)$$

$$\sum_{p \in OD} Larc_l^p \leq \sum_{f \in F} Cap_f x_{fl} \quad \forall l \in L, \quad (13e)$$

$$Darc_a \leq \tilde{\mu}_a \quad \forall a \in DA, \quad (13f)$$

$$Larc_l^p \leq \max_{f \in F} Cap_f \quad \forall l \in L, p \in OD, \quad (13g)$$

$$x \text{ binary, } (y, Larc, Darc, Harc) \geq 0. \quad (13h)$$

Yan and Tseng (2002) also consider ground-holding costs for the y -variables and passenger waiting costs for the $Harc$ -variables in the objective function, in addition to terms given in (13a) above. Constraints (13d) ensure flow balance in the passenger networks. Constraints (13e) enforce the capacity restriction on the number of passengers accepted on each leg $l \in L$. These are the very constraints that link the fleet-flow networks and the passenger-flow networks. Constraints (13f) and (13g) set upper bounds on the demand arcs and the delivery arcs as given by the projected demands and the maximum aircraft capacity, respectively. Notice that although the capacity constraints are implied by (13e), they are incorporated within the formulation for the sake of the proposed Lagrangian relaxation solution approach, as described next. Note that given a solution to this model, the fleet schedule would include only those legs l , $l \in L$, that have $\sum_{f \in F} x_{fl} = 1$, and would omit those that have $x_{fl} = 0 \quad \forall f \in F$. Moreover, both the fleet assignment and the passenger mix decisions would be at hand.

As a solution approach for this model, Yan and Tseng (2002) resort to Lagrangian relaxation where the Lagrangian multipliers are revised using a subgradient optimization method. In this approach, the constraints (13c) and (13e), along with the availability constraints, are accommodated within the objective function using non-negative Lagrangian multipliers. The objective value from the Lagrangian dual subproblem yields a lower bound for the original problem. An upper bound is found based on each subproblem solution by observing that the remaining constraints are separately related to the fleet-flow networks

and the passenger-flow networks: the balance constraints alone are for the fleet networks and the remaining are for the passenger networks. Hence, these two network-based sub-problems are solved separately. The obtained fleet solution is modified to sequentially satisfy constraints (13c) and the availability constraints by fixing variables heuristically, while maintaining network feasibility. Constraints (13e) are satisfied by subtracting passenger flows from the passenger network whenever no corresponding legs can be found in the fleet network, and filling vacancies in the assigned fleet having unsatisfied demands. The Lagrangian dual optimization is continued until either the gap between the upper and lower bounds are within an acceptable tolerance, or the number of iterations exceeds a prescribed limit.

This proposed heuristic is applied to an 11-station, 2-type, and 170-potential-flight network. The method is observed to converge to within a 2.5% gap within 5287 cpu seconds on a Pentium 200 computer. A better performance (2.4% gap and 2546 cpu seconds) is reported when also directly solving the LP relaxation first to obtain a lower bound for this particular problem.

5.2. A weekly fleet model

All the fleet assignment models reviewed above focus on the same-every-day fleet assignments (with the exception of Barnhart et al., 1998). The advantage of this approach is that assigning the same fleet types to the legs every day makes it easy to manage. However, flight schedules may vary over the days. For example, some legs are offered every other day. Moreover, demands on some legs may vary significantly over the different days of the week. Under these circumstances, a weekly assignment consideration may be preferred.

A daily fleet assignment approach can always be expanded to a weekly assignment by extending the time-lines of the network to cover an entire week. However, simply applying the daily fleet assignment approach cannot guarantee the same type of aircraft for the same flight leg flown at the same time (i.e., for a particular *flight number*). Since the latter is desirable from both the airline and passenger perspectives, additional modeling refinements are required to retain such a consistency in the assignments. This is the problem studied in Bélanger et al. (2004a,b). Their idea is that if a *dominant* fleet type is assigned to a flight number, discrepancies from this dominant type in the assignment can be reduced by penalizing differing assignments. Accordingly, they define a binary decision variable d_{fn} , which equals 1 if the dominant fleet type is f for flight number n , $n \in \Psi$, where Ψ is the set of flight numbers, and equals 0 otherwise. Also, let the variable p_{fl} denote whether or not the type assigned to leg l , $l \in L$, differs from the dominant type for its flight number, where the flight number for leg l is denoted by $n(l)$, and $n(l) \in \Psi$. This difference is penalized by a small number, γ , in the objective function. The following model is then proposed by Bélanger et al. (2004a,b):

Model 5.4. A weekly fleet assignment model with homogeneity

$$\text{Minimize} \quad \sum_{l \in L} \sum_{f \in F} (c_{fl} x_{fl} + \gamma p_{fl}) \quad (14a)$$

$$\text{subject to} \quad \text{cover (2b), balance (2c), and availability (2d), plus:} \quad (14b)$$

$$\sum_{f \in F} d_{fn} = 1 \quad \forall n \in \Psi, \quad (14c)$$

$$x_{fl} - d_{fn(l)} - p_{fl} \leq 0 \quad \forall f \in F, l \in L, \quad (14d)$$

$$(x, d) \text{ binary, } (y, p) \geq 0. \quad (14e)$$

The two new constraints (14c) and (14d) deal with the consistency in the fleet assignment. While (14c) ensures that only one dominant fleet type is assigned to each flight number, (14d) accounts for the number of assignments to leg l , $l \in L$, that differ from its dominant type. That is, whenever $x_{fl} = 1$ while $d_{fn(l)} = 0$, a penalty $p_{fl} = 1$ will be applied in the objective function.

Solving this model to optimality for an entire week is an arduous task. As a result, Bélanger et al. (2004a,b) propose two simple heuristics. The first heuristic uses a depth-first branch-and-bound approach and stops once a first feasible solution is discovered. This heuristic is quite time consuming and solves a 2000-leg problem in around 20 hours on a SUN Ultra-10/440 workstation. Alternatively, a two-phase heuristic is proposed, which can significantly reduce the computational effort without significantly sacrificing the solution quality. In this heuristic, the first phase considers only a small subset of the legs, to determine the dominant type for each flight number. The second phase then uses these dominant types as given, and solves the resulting problem for the entire week of flights. Numerical experiments demonstrate that to achieve the aforementioned consistency in the fleet assignment via this modeling and solution approach, the loss in overall profits is no more than 1.7% for several sample schedules involving up to 2000 flights.

5.3. Combined through-fleet assignment model

Recall, from Section 2, that a through-flight refers to two or more legs that are desirable to be flown by the same aircraft. Through-flights are attractive to those passengers flying multiple legs between their origins and destinations, because even though the aircraft stops at intermediate stations, the passengers can stay onboard until their final destinations. An extra revenue can be gained by forming through-flights, since passengers usually pay a higher price for this convenience.

Researchers have proposed several approaches to include the consideration of through-flights within the FAM. A typical approach solves the FAM, and then determines the through-flights based on this solution, considering the potential revenue benefits (Gopalan and Talluri, 1998). However, this sequential approach is limited in its ability to achieve higher revenue solutions, because the through-flights are confined only to legs that are assigned to the same fleet types in the FAM. The string-based model proposed by Barnhart et al. (1998) (see Section 4.2) can also be used to determine the through-flights, since a string corresponds to a sequence of legs flown by one aircraft. More recently, Ahuja et al. (2002) have developed a *Very Large-Scale Neighborhood (VLSN) Search Algorithm* for a Combined Through-Fleet Assignment Model (ctFAM). This approach starts from a feasible solution obtained by solving the FAM, not necessarily to optimality, and then determines the through-flights sequentially. The algorithm is based on generating neighbor solutions, which differ in their fleet assignments from the current solution for only two aircraft types. If a neighbor solution leads to an improvement over the current fleet solution, then the corresponding aircraft types are swapped. This neighborhood search algorithm, as well as a tabu search algorithm, are tested on four problems obtained from United Airlines. Using the neighborhood search algorithm with and without maintenance considerations yields an additional annual revenue of \$26.86 million and \$17.15 million, respectively, over solving the IP model for 30 minutes. Using tabu search generates relatively higher revenues, but at the expense of much longer computational times.

Based on this work, Ahuja et al. (2003) further solve the multi-criteria ctFAM to generate *Pareto-optimal solutions*, i.e., solutions that are not dominated by any other solution with respect to all criteria, with the dominance with respect to at least one criterion being strict. In their implementation, two particular criteria that pertain to ground manpower scheduling and crew scheduling are used. Two standard approaches are used for the multi-criteria search, based respectively on either converting the multiple objective functions into a single objective function by using appropriate weight parameters, or optimizing with respect to the two objectives sequentially, while restricting the first objective value to not exceed a pre-specified value in the second run. Numerical results suggest that the latter approach yields better solutions than the former method, and using tabu search yields slightly better non-dominated solutions than using the neighborhood search algorithm.

5.4. Non-optimization based fleet solutions

Aside from the neighborhood search algorithm, some researchers have investigated the effectiveness of more randomized local search procedures. For example, a greedy randomized adaptive search procedure (GRASP)

and simulated annealing (SA) have been studied by Sosnowska (2000). Both GRASP and SA start with randomly generated initial solutions, which consist of a rotation cycle for each aircraft, and perform a search that may lead to a move from the current solution to one of its neighbors. A neighbor is created by a *swap operation*. The swap operation selects a pair of aircraft, determines a common airport in their rotation cycles, and switches the portions of the two rotation cycles following the common airport, if it is feasible to do so. In particular, the balance constraints and time constraints need to be satisfied after the swap, i.e., the first portion of one of the rotation cycles should terminate before the start of the second portion of the other rotation cycle. In the GRASP algorithm, an inner loop and an outer loop are performed. The inner loop is a *local optimization phase*, which randomly selects a fixed number of pairs of rotation cycles and performs swap operations. The swap that improves the cost most significantly is stored. This is then passed to the outer loop, a *construction phase*, which fixes the best several swaps from the current and the previous iterations, and reassigns the remaining legs randomly to the other aircraft. Unlike GRASP, which retains only the improving solutions during the neighborhood searches, the SA accepts not only improving solutions but also a proportion of deteriorating ones, with the hope of not getting trapped at a local optimum. The expected demand, along with the cost for one hour of flight time, are used to compute the expected profit. Results show that the GRASP approach performs better when the number of swaps per local search and the percentage of rotation cycles fixed at the outer loop are relatively higher. The results are based on test problems having about 20 aircraft and 3000 flights. However, no computational times or estimated optimality gaps are reported in the paper.

6. Dynamic re-fleeting mechanisms based on updated demands

The current airline practice is to assign aircraft capacity to scheduled flights well in advance of departures. This is due to the need to generate the crew schedules 8–12 weeks in advance of the departure times, as dictated by typical union contracts and government regulations, and also, because fleet decisions provide a major input to other airline processes, such as revenue management. At such an early stage, however, the high uncertainty of demand poses a major impediment for airlines to best match aircraft capacities with the final demand. On the other hand, the accuracy of demand forecasts improve markedly over time. Thus, it becomes advantageous to *delay* (or postpone) irrevocable fleet decisions to as great a future extent as possible, and to inject sufficient *flexibility* in the prescribed decisions so as to facilitate the ability to make future revisions within the operational constraints.

In this section, we review the research on dynamic re-fleeting as driven by updated demand information. Demand driven re-fleeting decisions require frequent updates to reflect rapid market changes. Other types of changes, such as operational disturbances and changes in maintenance requirements, also frequently require fleet assignments to be updated, but in this case, a solution is usually needed in a much shorter time-frame. Thus, this type of re-fleeting focuses on the perturbed constraints, and seeks a one-time correction to the solution at hand, usually via some fast heuristic approach. We refer the interested reader for such re-fleeting approaches to the recent articles by Jarrah et al. (2000), Thengvall et al. (2000), Rosenberger et al. (2003), as well as to the relevant references cited therein. For the remainder of this section, we will focus on the demand driven re-fleeting issue.

Any re-fleeting decision that is accompanied by changes in the current crew assignments is very difficult and expensive to implement. As a result, it becomes advantageous to confine the re-assignment to lie within each family of aircraft and to the legs assigned to this particular family. (Recall, from Section 2, that a crew is qualified to fly any type within any given family.) Hence, the re-assignment of an aircraft within a given family would only possibly change the aircraft type that was originally assigned to some crew to another type having a different capacity, but yet serviceable by the same crew.

Berge and Hopperstad (1993) were one of the first researchers to propose a demand driven re-fleeting approach, referred to as the *demand driven dispatch* (D^3), which deals with the dynamic re-assignment of

aircraft to legs within each family, utilizing demand forecasts that improve as departures approach. The sample problems they used to test the effectiveness of their D^3 approach focused on a single week's flights. The fleet decisions were revised multiple times prior to flight departures, while the demand forecasts were updated based on the actual bookings. At the departure time of each flight, the realized demand served and the associated revenue outcome were recorded in order to evaluate the performance of the D^3 approach. Observe that the D^3 method is applied to only one family at a time, and that all the aircraft types in the same family are assumed to have the same flight- and turn-times. Consequently, the re-fleet network can adopt a single network for all types in each family. The resulting formulation is therefore a variation of the basic FAP that is defined on a time-space network similar to that described in Section 3.2.

Instead of solving the resulting IP to optimality, Berge and Hopperstad (1993) proposed two heuristics to find re-fleet solutions in significantly shorter times. The first approach is based on the observation that if there are only two fleet types, then using the cost (or profit) differences of these two types as the objective coefficients reduces the problem to a single commodity minimum cost flow problem, which then needs to be solved for only one type. The resulting solution and its complement constitute the assignments for the two types. Using the essence of this idea, a heuristic is developed in which fleet types are sorted in order of increasing capacities. At each iteration, the first two types in the list are used to solve a single commodity flow problem using a successive shortest path method, and the smaller type is removed from the list, and the process is repeated. The second heuristic starts by constructing a feasible assignment, and attempts to improve this solution by finding pairs of paths that share the same origin and destination and that are assigned to two types such that switching the assignments of these two types on these paths improves the total profit.

Their case studies, based on actual airline networks, revealed a 1–5% potential improvement in the operating profit from the application of the D^3 approach, realized mainly due to spill avoidance and a reduction in the number of larger airplanes used. The proposed heuristics on test problems having about 2000 weekly flights and 2–3 types with a total of about 50 aircraft yielded solutions within 99.9% of optimality. Continental Airlines (Pastor, 1999) has also performed limited experiments to test the benefits of the D^3 concept, and has reported significant gains as a result of utilizing this flexibility in the system.

The second heuristic by Berge and Hopperstad (1993) can find swapping opportunities when a pair of aircraft of two different types are assigned to two paths that share the same origin and destination, but it would fail if the swapping opportunity involves multiple pairs of aircraft. Talluri (1996) improves upon this heuristic by proposing an algorithm that is guaranteed to find a same-day swap opportunity for two fleet types, if one exists. His algorithm generates a swap having the fewest number of changes. For a network having n nodes and m arcs, Talluri's algorithm has a running time of $O(m)$ compared to the running time of $O(mn)$ for Berge and Hopperstad's method.

From a different perspective, Bish et al. (2004) study the benefits of several demand driven swapping (DDS) mechanisms characterized in terms of their *timing* (when the swapping decision should be made) and *frequency* (how often the swapping decision should be revised). As with any re-fleet decision, swapping aircraft early in time will cause less disturbance to revenue management and operations, but will be based on more uncertain demand. On the other hand, allowing revisions to the initial swapping decision when demand uncertainty is reduced will benefit from the use of improved demand information, thus reducing the possibility of swaps with loss, but will disrupt operations and revenue management more. In addition, if the swapping decision is delayed too much, then some customers may have already been rejected due to capacity restrictions. Note that the proposed DDS approach is intended to be implemented at close proximity to departures (i.e., six weeks) so as to incorporate the updated demand forecasts into the fleet decisions. Accordingly, Bish et al. (2004) consider swapping aircraft that are assigned to simple loops within similar time-frames where each such loop consists of a round-trip that originates and terminates at a common airport. Using both analytical models as well as numerical integration and simulation studies, they determine conditions under which each of the different DDS strategies is effective. The proposed

methodology is also tested using data obtained from United Airlines. The results reveal that implementing a DDS strategy can significantly improve an airline's revenue.

Since the re-fleeting problem is solved for one family at a time, more accurate and detailed path-level (as opposed to leg-based) demand forecasts that are obtained as departures approach can be incorporated into the re-fleeting model. This is the problem studied in Sherali et al. (in press). Although the resulting formulation is somewhat similar to the initial FAP model with path-based demands, which is presented in Section 5.1, this problem is solved on a relatively smaller scale by considering re-fleeting decisions only for the legs that are assigned to a particular family, while keeping the fleet assignments for the other families fixed. The legs that are assigned to the other families, however, still need to be included in the formulation. This is because the fleet types assigned to those legs can still restrict the number of passengers that can be accepted on paths that contain both these legs and those pertaining to the particular family that is to be re-fleeted. This is considered in constraints (15d) in the following model. Notationally, for this model, the index k is used to denote the designated family k under present consideration from the entire family set K , $CapAss_l$ is the fixed capacity for a leg l assigned to other families, $\forall l \in L \setminus L_k$, and q_i is the number of passengers accepted on path $i \in \Pi$, without considering the recapture effect.

Model 6.1. *Integrated path-based re-fleeting model for family k :*

$$\text{Minimize} \quad \sum_{l \in L_k} \sum_{f \in F_k} c_{fl} x_{fl} - \sum_{i \in \Pi} fare_i q_i \quad (15a)$$

$$\text{subject to} \quad \text{cover (2b), balance (2c), availability (2d), plus:} \quad (15b)$$

$$\sum_{i \in \Pi(l)} q_i \leq \sum_{f \in F_k} Cap_f x_{fl} \quad \forall l \in L_k, \quad (15c)$$

$$\sum_{i \in \Pi(l)} q_i \leq CapAss_l \quad \forall l \in L \setminus L_k, \quad (15d)$$

$$q_i \leq \mu_i \quad \forall i \in \Pi, \quad (15e)$$

$$x \text{ binary, } (y, q) \geq 0. \quad (15f)$$

Although the restriction of the problem to a single family reduces the number of leg-related variables in the formulation, the path-based considerations still render this problem as very challenging, while fast computational times are needed due to the relative proximity of the decision period to departure. Consequently, Sherali et al. (in press) perform a polyhedral analysis of Model 6.1 and design solution approaches using several reformulation and partial convex hull construction mechanisms, along with various classes of valid inequalities to tighten the model representation. Valid inequalities are derived from selected constraints and surrogates of constraints to achieve a tighter LP representation while avoiding excess model size growth. An LP-based separation routine is designed to search for paths $i \in \Pi$ that can be selected to advantageously construct facets of the convex hull of solutions feasible to a substructure of the model. The entire convex hull of this substructure is also directly constructed in a disaggregated higher dimensional space.

Properly composing these strategies greatly enhances the overall effectiveness of the model, inducing its LP relaxation to yield a relatively greater number of variables that automatically turn out to be binary-valued. The developed procedure is able to solve moderately-sized problem instances having 18,196 to 33,105 paths, 800–1600 flights, including 300 to 396 flights from the designated family, to provable optimality within a reasonable tolerance (5% of optimality). In addition, this polyhedral analysis-based approach was also applied to solve a real problem posed by United Airlines, having 5098 flight legs (with 459 legs in the family) and 64,247 paths, within 20 cpu minutes of computing time on a Sun-Blade-1000 (UltraSP-ARC-III) workstation having 512 MB RAM and a cpu speed of 750 MHz. The tight relaxations also facilitated the design of an effective heuristic for solving relatively larger sized problems. In this procedure, some

of the binary variables were sequentially fixed in value via the solution of a series of LP relaxations in order to reduce the size of the problem, which was then fed into an MIP solver (CPLEX 8.0). The results indicated that the solution effort required was significantly reduced, while the quality of the solution was only mildly degraded.

7. Future research directions

To tackle more realistic versions of the fleet assignment problems that are encountered in practice, numerous extensions can be included. These include the consideration of path-based demands, network and recapture effects, through-flights, maintenance issues, etc., as variously discussed in this paper, and also other market and/or airline dependent requirements. One particular example is by [Bélanger et al. \(2004a,b\)](#), who aim to insert sufficient time between departure times of consecutive flights serving the same OD pair, so that these flights will not compete with each other. For this purpose, they incorporate time-windows, spacing constraints, and time dependent revenues into FAM. Airline operational decisions are usually made sequentially in the order of scheduling, FAM, routing, and crew scheduling. Each module ought to be solved so as to provide a maximum level of flexibility for the subsequent modules so that the total cost of the entire operational chain can be reduced. Changing the decision sequence for some of the modules may also open a different perspective of modeling the entire process. This type of experimentation includes [Klabjan et al. \(2002\)](#), who solve crew scheduling before routing, and use plane-count constraints to maintain feasibility for routing decisions. Increasing module flexibility leads to the ultimate goal of integrating these processes. We have included integrated models of FAM with scheduling, routing, and crew considerations in Section 4. Other integrated type of airline operational models include, for example, the integration of routing and crew scheduling by [Cordeau et al. \(2001\)](#), [Cohn and Barnhart \(2002\)](#) and [Mercier et al. \(2005\)](#), and the integration of scheduling and routing for charter airlines by [Erdmann et al. \(2001\)](#). Following the trend of integrating the interacting airline operations, additional global integration investigations naturally offer future research directions. Looking forward, airline operations, airline yield management and pricing decisions should all be integrated to achieve a synchronized system ([Jacobs et al., 2000](#)). However, the problem size will grow at each further integrative step, and will call for more efficient solution techniques.

One major limitation in the proposed models is the estimation of the model parameters, such as flight fares, demands, recapture rates, etc. The accuracy of the solution highly depends on these values. In the current airline market, the demands change more drastically. Hence, robust models are imperative so as to obtain solutions that are good in not only the ideal or expected instances, but even in the unexpected cases. Current robust airline operational models mostly deal with crew scheduling (for instance, [Schaefer \(2000\)](#), [Yen and Birge \(in press\)](#), [Klabjan et al. \(2001\)](#), [Schaefer et al. \(2001\)](#), and [Ehrgott and Ryan \(2002\)](#), as detailed in [Rosenberger et al. \(2004\)](#)). Robustness concepts need to be included into the models for the initial fleet assignment process, and supporting approaches need to be developed to deal with model parameter updates during the re-fleet assignment stage.

Another major issue in the airline planning processes is operations recovery. [Rosenberger et al. \(2002\)](#) have presented a stochastic model and have implemented it via a simulation program, *SimAir*, for prescribing airline daily operations, and for evaluating the performance of plans and recovery operational policies. Fleet assignment is part of this overall evaluation program. Many other models also study airline operations recovery phenomena under perturbation, though most focus on schedule recovery ([Jarrah et al., 1993](#); [Luo and Yu, 1997](#); [Thengvall et al., 2000](#); [Stojković et al., 2002](#)); and on crew recovery ([Stojković et al., 1998](#); [Yu et al., 2003](#)). The fleet assignment recovery models include [Jarrah et al. \(2000\)](#). Nonetheless, a deeper investigation of alternative scenarios along with the development of effective solution approaches deserves further attention.

Another major research direction stems from the fact that current re-fleeting approaches hardly address any feedback process to the initial fleeting stage. Clearly, the initial fleeting decisions and the re-fleeting processes are highly dependent on each other. The initial fleeting decisions greatly constrain the downstream re-fleeting possibilities, in that it determines the flexibility with which the re-fleeting process can exploit (for example, most re-fleeting is limited to aircraft of the same family due to crew concerns; see Berge and Hopferstad (1993), Bish et al. (2004), and Sherali et al. (in press)). On the other hand, the subsequent re-fleeting prospects for performing revisions to the fleeting solution need to be considered upfront in the initial fleeting decision in order to retain a sufficient degree of flexibility in the system. Consequently, the initial fleeting and the subsequent re-fleeting processes can be considered analogous to supply (capacity) management that is implemented in other industries. Thus, an important and challenging future research direction would be to explore the interactions between these problems, and thereby, to coordinate them. Future research in this area is intimately related with the robustness, supply (capacity) flexibility, and decision postponement issues. The robustness issue was discussed earlier. Therefore, we conclude this paper with a discussion on supply flexibility and decision postponement strategies. The benefits of supply flexibility and decision postponement strategies have been extensively researched within the domain of manufacturing and other service industries (such as retailing); see, for instance, Bish (in press), Tayur et al. (2000), and Van Mieghem (2003) for related literature on flexible capacity management and decision postponement strategies within the context of supply chain and operations management. The implementation of these ideas have resulted in major success stories for several companies in the manufacturing and retailing industries (see the many references in Tayur et al. (2000)). In contrast, the idea of a *systematic* supply management in the airline industry has neither been much researched nor implemented. Although the different characteristics and constraints of airline companies will warrant a specialized investigation on the utilization of supply flexibilities, ideas and inspirations can still be gleaned from other industries that have used these techniques successfully, such as the manufacturing and retailing industries. Thus, we believe that further work that considers these synergies accruing from different application domains will be a promising avenue for future research.

Acknowledgement

This research is supported in part by the National Science Foundation under Grant Numbers DMI 0094462, DMI 0010032, and DMI 0245643.

References

- Abara, J., 1989. Applying integer linear programming to the fleet assignment problem. *Interfaces* 19 (4), 20–28.
- Ahuja, R.K., Goodstein, J., Mukherjee, A., Orlin, J.B., Sharma, D., 2002. A very large-scale neighborhood search algorithm for the combined through and fleet assignment model. Working paper, MIT Sloan School of Management, 4388-01, Cambridge, MA. Downloadable from website <<http://ideas.repec.org/p/mit/sloanp/1796.html>>.
- Ahuja, R.K., Liu, J., Goodstein, J., Mukherjee, A., Orlin, J.B., Sharma, D., 2003. Solving multi-criteria through-fleet assignment models. In: Ciriani, T.A., Fasano, G., Gliozzi, S., Tadei, R. (Eds.), *Operations Research in Space and Air*. Kluwer Academic Publishers, Dordrecht, pp. 233–256.
- Ball, M.O., 2004. Introduction to the special issue on aviation operations research: Commemorating 100 years of aviation. *Transportation Science* 37, 366–367.
- Barnhart, C., Belababa, P., Odoni, A.R., 2004. Applications of operations research in the air transport industry. *Transportation Science* 37, 368–391.
- Barnhart, C., Boland, N.L., Clarke, L.W., Johnson, E.L., Nemhauser, G.L., Shenoi, R.G., 1998. Flight string models for aircraft fleeting and routing. *Transportation Science* 32, 208–220.
- Barnhart, C., Cohn, A., Johnson, E.L., Klabjan, D., Nemhauser, G.L., Vance, P., 2003. Airline crew scheduling. In: Hall, R.W. (Ed.), *Handbook of Transportation Science*. Kluwer Academic Publishers, Boston, MA, pp. 517–560.

- Barnhart, C., Kniker, T.S., Lohatepanont, M., 2002. Itinerary-based airline fleet assignment. *Transportation Science* 36, 199–217.
- Bélanger, N., Desaulniers, G., Soumis, F., Desrosiers, J., Lavigne, J., 2004a. Airline fleet assignment with homogeneity. Working paper, École Polytechnique and GERAD, Montréal, Québec, Canada.
- Bélanger, N., Desaulniers, G., Soumis, F., Desrosiers, J., 2004b. Periodic airline fleet assignment with time windows, spacing constraints, and time dependent revenues. Submitted to *Transportation Science*.
- Berge, M.E., Hopperstad, C.A., 1993. Demand driven dispatch: A method for dynamic aircraft capacity assignment, models and algorithms. *Operations Research* 41, 153–168.
- Bish, E.K., Optimal investment strategies for flexible resources, considering pricing. In: Geunes, J., Pardalos, P.M. (Eds.), *Supply Chain Optimization*. Kluwer Academic Publishers, in press.
- Bish, E.K., Suwandechochai, R., Bish, D.R., 2004. Strategies for managing the flexible capacity in the airline industry. *Naval Research Logistics* 51, 654–685.
- Clarke, M., Smith, B., 2004. Impact of operations research on the evolution of the airline industry. *Journal of Aircraft* 41, 62–72.
- Clarke, L.W., Hane, C.A., Johnson, E.L., Nemhauser, G.L., 1996. Maintenance and crew considerations in fleet assignment. *Transportation Science* 30, 249–260.
- Cohn, A., Barnhart, C., 2002. Improving crew scheduling by incorporating key maintenance routing decisions. *Operations Research* 51, 387–396.
- Cordeau, J., Stojković, G., Soumis, F., Desrosiers, J., 2001. Benders decomposition for simultaneous aircraft routing and crew scheduling. *Transportation Science* 35, 375–388.
- Desaulniers, G., Desrosiers, J., Dumas, Y., Solomon, M.M., Soumis, F., 1997. Daily aircraft routing and scheduling. *Management Science* 43, 841–855.
- Desrosiers, J., Pelletier, P., Soumis, F., 1983. Plus court chemin avec contraintes d'horaires. *RAIRO* 17, 357–377 (in French).
- Dror, M., Trudeau, P., Ladany, S.P., 1988. Network models for seat allocation on flights. *Transportation Research* 22b, 239–250.
- Ehr Gott, M., Ryan, D.M., 2002. Constructing robust crew schedules with bicriteria optimization. *Journal of Multi-Criteria Decision Analysis* 11, 139–150.
- Erdmann, A., Nolte, A., Noltmeier, A., Schrader, R., 2001. Modeling and solving the airline schedule generation problem. *Annals of Operations Research* 107, 117–142.
- Farkas, A., 1995. The influence of network effects and yield management on airline fleet assignment decisions. Ph.D. Dissertation, Massachusetts Institute of Technology, Cambridge, MA.
- Glover, F., Glover, R., Lorenzo, J., McMillan, C., 1982. The passenger mix problem in the scheduled airlines. *Interfaces* 12, 73–79.
- Gopalan, R., Talluri, K.T., 1998. Mathematical models in airline schedule planning: A survey. *Annals of Operations Research* 76, 155–185.
- Gu, Z., Johnson, E.L., Nemhauser, G.L., Wang, Y., 1994. Some properties of the fleet assignment problem. *Operations Research Letters* 15, 59–71.
- Hane, C.A., Barnhart, C., Johnson, E.L., Marsten, R.E., Nemhauser, G.L., Sigismond, G., 1995. The fleet assignment problem: Solving a large-scale integer program. *Mathematical Programming* 70, 211–232.
- Jacobs, T.L., Smith, B.C., Johnson, E., 1999. O&D FAM: Incorporating passenger flows into the fleet planning process. In: *Proceedings of the AGIFORS Symposium*, New Orleans, LA, pp. 128–161.
- Jacobs, T.L., Ratliff, R.M., Smith, B.C., 2000. Soaring with synchronized systems. *OR/MS Today* 37, 36–44.
- Jarrah, A.I., Yu, G., Krishnamurthy, N., Rakshit, A., 1993. A decision support framework for airline cancellations and delays. *Transportation Science* 27, 266–280.
- Jarrah, A.I., Goodstein, J., Narasimhan, R., 2000. An efficient airline re-fleet model for the incremental modification of planned fleet assignments. *Transportation Science* 34, 349–363.
- Klabjan, D. Large-scale models in the airline industry. In: Desaulniers, G., Desrosiers, J., Solomon, M.M. (Eds.), *Column Generation*, Kluwer Academic Publishers, in press.
- Klabjan, D., Schaefer, A.J., Johnson, E.L., Kleywegt, A.J., Nemhauser, G.L., 2001. Robust airline crew scheduling. In: *Proceedings of the Triennial Symposium on Transportation Analysis (TRISTAN IV)*, São Miguel, Azores Islands, Portugal, pp. 275–280.
- Klabjan, D., Johnson, E.L., Nemhauser, G.L., Gelman, E., Ramaswamy, S., 2002. Airline crew scheduling with time windows and plane-count constraints. *Transportation Science* 36, 337–348.
- Kniker, T.S., 1998. Itinerary-based airline fleet assignment. Ph.D. Dissertation, Massachusetts Institute of Technology, Cambridge, MA.
- Kohl, N., Madsen, O.B.G., 1997. An optimization algorithm for the vehicle routing problem with time windows based on Lagrangian relaxation. *Operations Research* 45, 395–406.
- Levin, A., 1971. Scheduling and fleet routing models for transportation systems. *Transportation Science* 5, 232–255.
- Lohatepanont, M., Barnhart, C., 2004. Airline schedule planning: Integrated models and algorithms for schedule design and fleet assignment. *Transportation Science* 38, 19–32.
- Luo, S., Yu, G., 1997. On the airline schedule perturbation problem caused by the ground delay program. *Transportation Science* 31, 298–311.

- Mercier, A., Cordeau, J.F., Soumis, F., 2005. A computational study of Benders decomposition for the integrated aircraft routing and crew scheduling problem. *Computers and Operations Research* 32 (6), 1355–1664.
- Pastor, J., 1999. Demand driven dispatch: A stochastic approach. In: *Proceedings of the AGIFORS Symposium*, New Orleans, LA, pp. 162–184.
- Phillips, B.L., Boyd, D.W., Grossman, T.A., 1991. An algorithm for calculating consistent itinerary flows. *Transportation Science* 25, 225–239.
- Rexing, B., Barnhart, C., Kniker, T.S., Jarrah, T.A., Krishnamurthy, N., 2000. Airline fleet assignment with time windows. *Transportation Science* 34, 1–20.
- Rosenberger, J.M., Schaefer, A.J., Goldsman, D., Johnson, E.L., Kleywegt, A.J., Nemhauser, G.L., 2002. A stochastic model of airline operations. *Transportation Science* 36, 357–377.
- Rosenberger, J.M., Johnson, E.L., Nemhauser, G.L., 2003. Rerouting aircraft for airline recovery. *Transportation Science* 37, 408–421.
- Rosenberger, J.M., Johnson, E.L., Nemhauser, G.L., 2004. A robust fleet-assignment model with hub isolation and short cycles. *Transportation Science* 38, 357–368.
- Rushmeier, R.A., Kontogiorgis, S.A., 1997. Advances in the optimization of airline fleet assignment. *Transportation Science* 31, 159–169.
- Schaefer, A.J., 2000. Airline crew scheduling under uncertainty. Ph.D. Dissertation, Georgia Institute of Technology, Atlanta, GA.
- Schaefer, A.J., Johnson, E.L., Kleywegt, A.J., Nemhauser, G.L., 2001. Airline crew scheduling under uncertainty. Technical Report TLI/LEC-01-01, Georgia Institute of Technology, Atlanta, GA, Downloadable from website <<http://gunther.smeal.psu.edu/14004.html>>.
- Sherali, H.D., Bish, E.K., Zhu, X. Polyhedral analysis and algorithms for a demand driven re-fleet model for aircraft assignment. *Transportation Science*, in press.
- Sosnowska, D., 2000. Optimization of a simplified fleet assignment problem with metaheuristics: Simulated annealing and GRASP. In: Pardalos, P.M. (Ed.), *Approximation and Complexity in Numerical Optimization: Continuous and Discrete Problems*. Kluwer Academic Publishers, Dordrecht, The Netherlands, pp. 477–488.
- Stojković, M., Soumis, F., Desrosiers, J., 1998. The operational airline crew scheduling problem. *Transportation Science* 32, 232–245.
- Stojković, G., Soumis, F., Desrosiers, J., Solomon, M.M., 2002. An optimization model for a real-time flight scheduling problem. *Transportation Research* 36a, 779–788.
- Subramanian, R., Scheff, R.P., Quillinan, J.D., Wiper, D.S., Marsten, R.E., 1994. Coldstart: Fleet assignment at delta airlines. *Interfaces* 24 (1), 104–120.
- Talluri, T.K., 1996. Swapping applications in a daily airline fleet assignment. *Transportation Science* 30, 237–248.
- Talluri, T.K., Van Ryzin, G., 1999. A randomized linear programming method for computing network bid prices. *Transportation Science* 33, 207–216.
- Tayur, S., Ganeshan, R., Magazine, M., 2000. *Quantitative Models for Supply Chain Management*. Kluwer Academic Publishers.
- Thengvall, B.G., Bard, J.F., Yu, G., 2000. Balancing user preferences for aircraft schedule recovery. *IEEE Transactions* 32, 181–193.
- Van Mieghem, J.A., 2003. Capacity management, investment and hedging: Review and recent developments. *Manufacturing & Service Operations Management* 5, 269–302.
- Yan, S., Tseng, C.-H., 2002. A passenger demand model for airline flight scheduling and fleet routing. *Computers and Operations Research* 29, 1559–1581.
- Yen, J., Birge, J. A stochastic programming approach to the airline crew scheduling problem. *Transportation Science*, in press.
- Yu, G., Yang, J., 1998. Optimization applications in the airline industry. In: Du, D.Z., Pardalos, P.M. (Eds.), *Handbook of Combinatorial Optimization*. Kluwer Academic Publishers, Norwell, MA, pp. 635–726.
- Yu, G., Arguello, M., Song, G., McCowan, S.M., White, A., 2003. A new era for crew recovery at Continental Airlines. *Interfaces* 33 (1), 5–22.