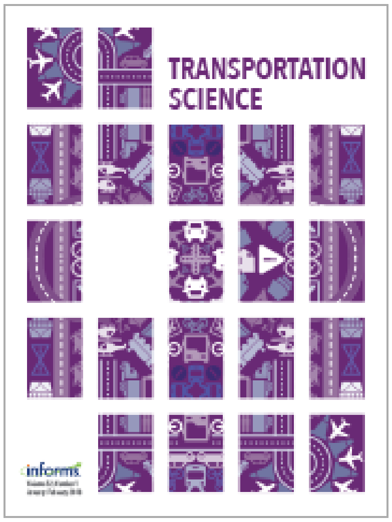

Transportation Science

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

Flight String Models for Aircraft Fleeting and Routing

Cynthia Barnhart, Natashia L. Boland, Lloyd W. Clarke, Ellis L. Johnson, George L. Nemhauser, Rajesh G. Shenoi,

Flight String Models for Aircraft Fleeting and Routing[1]

CYNTHIA BARNHART

Center for Transportation Studies, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139

NATASHIA L. BOLAND

Department of Mathematics, University of Melbourne, Parkville VIC 3052 Australia

LLOYD W. CLARKE, ELLIS L. JOHNSON and GEORGE L. NEMHAUSER

School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, Georgia 30332-0205

RAJESH G. SHENOI

McKinsey & Company, 2 Houston Center, Suite 3500, Houston, Texas 77010

Given a schedule of flight legs to be flown by an airline, the fleet assignment *problem is to determine the minimum cost assignment of flights to aircraft types, called fleets, such that each scheduled flight is assigned to exactly one fleet, and the resulting assignment is feasible to fly given a limited number of aircraft in each fleet. Then the airline must determine a sequence of flights, or routes, to be flown by individual aircraft such that assigned flights are included in exactly one route, and all aircraft can be maintained as necessary. This is referred to as the* aircraft routing *problem. In this paper, we present a single model and solution approach to solve simultaneously the fleet assignment and aircraft routing problems. Our approach is robust in that it can capture costs associated with aircraft connections and complicating constraints such as maintenance requirements. By setting the number of fleets to one, our approach can be used to solve the aircraft routing problem alone. We show how to extend our model and solution approach to solve aircraft routing problems with additional constraints requiring equal aircraft utilization. With data provided by airlines, we provide computational results for the combined fleet assignment and aircraft routing problems without equal utilization requirements and for aircraft routing problems requiring equal aircraft utilization.*

After creating a schedule defining the departure and arrival times for each flight leg to be flown, an airline must decide what equipment type, or *fleet*, should be assigned to each flight. This is referred to as the *fleet assignment* problem. Then, given an assignment of flights to fleets, the airline must determine a sequence of flights, or routes, to be flown by individual aircraft such that assigned flights are included in exactly one route, and each aircraft visits its maintenance stations at regular intervals. This is referred to as the *aircraft routing* problem.

In this paper, we present a string-based model and a branch-and-price solution approach to solve both the fleet assignment and aircraft routing problems. We define a *string* to be a sequence of connected flights that begins and ends at maintenance stations, satisfies flow balance (that is, a flight arriving at station j is followed by a flight departing station j) and is maintenance feasible. A string is

[1]Accepted by Gilbert Laporte.

Transportation Science
Vol. 32, No. 3, August 1998

0041-1655/98/3203-0208 $01.25

maintenance feasible if it satisfies all Federal Aviation Administration (FAA) and carrier-specified maintenance requirements.

Our modeling and solution approach is robust in that it can capture complicating constraints such as maintenance requirements or aircraft utilization restrictions. The drawback to our model is that flight schedules with hundreds of flights have millions of strings. In the following sections, we detail our modeling approach and describe an optimization-based solution approach that overcomes this drawback.

FLEET ASSIGNMENT

THE INPUTS FOR THE fleet assignment problem include a schedule of flight legs, a set of aircraft of various equipment types, called *fleets*, the cost of flying each flight leg by fleet type, *maintenance* constraints reflecting the routine maintenance requirements of the FAA, and turn time restrictions at each station by fleet type, which specify the minimum time needed after the arrival of an aircraft to prepare the plane for its next departure. The flight schedule is balanced with the number of flights into any location equaling the number out and periodic; that is, the set of flights repeats every T time units, for some value T. The solution to the fleet assignment problem assigns each flight to exactly one fleet, maintains balance for each fleet, and uses no more aircraft of a given fleet than are available.

This assignment is made to minimize costs, where cost is defined as the sum of aircraft operating costs and opportunity costs of losing *spilled* passengers due to excess demand. Spill costs are a function of both demand for a flight and aircraft capacity.

The fleet assignment problem has been well studied (ABARA, 1989; BALL, HOFFMAN, and RUSHMEIR, 1996; CLARKE et al., 1996; DESAULNIERS et al., 1994b; and HANE et al., 1995). In each of these cases, except Ball, Hoffman, and Rushmeir (1996) and Desaulniers et al. (1994b), the solution to the fleet assignment problem may not yield a feasible solution to the aircraft routing problem because individual aircraft are not considered. As a result, maintenance requirements cannot be modeled exactly. Instead, *aggregate maintenance* constraints are included that require a minimum number of *maintenance opportunities*. A maintenance opportunity exists if an aircraft is at a maintenance station for a sufficient period of time. These aggregate constraints are approximate because they do not guarantee that the maintenance opportunities are distributed equally among the individual aircraft. One aircraft may have more maintenance opportunities than it needs whereas another may have none.

Another shortcoming of conventional fleet assignment models is that *through revenues* are not captured. A sequence of flights flown by one aircraft, where there is demand from the origin of the first flight to the destination of the last one, is called a *through*. Throughs may result in extra revenue because passengers are willing to pay a premium to stay on the same aircraft rather than make a connection at an airport (GOPALAN and TALLURI, 1998a). Because conventional fleet assignment models do not determine the individual assignments of flights to aircraft, actual connections for aircraft are not determined and consequently, through revenues are ignored.

Our string-based model and branch-and-price solution approach to solve the fleet assignment problem guarantees satisfaction of maintenance requirements and also includes through-revenues.

AIRCRAFT ROUTING

Given a balanced, periodic flight schedule for a *single* fleet, the *aircraft routing* problem is to find a minimum-cost set of aircraft routings. Because only one fleet is involved, the costs to be minimized include the maintenance costs and the negative through revenues associated with the flight connections of the aircraft in that fleet. The routings must satisfy *flight coverage* constraints requiring each flight assigned to the fleet to be contained in exactly one route, *fleet count* constraints limiting each aircraft to be assigned to at most one route, and *maintenance* constraints requiring each aircraft to visit maintenance stations at least as frequently as required by the FAA.

The FAA mandates four main categories of safety checks on airlines, varying in scope, duration, and frequency (CLARKE et al., 1997). The maintenance checks modeled in the routing problem are the A checks, requiring each aircraft to be maintained for every 60 hours of flying. Airlines typically enforce more stringent maintenance requirements and require an A check every 40–45 hours of flying, with the maximum time between checks restricted to three to four calendar days.

Previous work on the aircraft routing problem is reported in Clarke et al. (1997), FEO and BARD (1989), Gopalan and Talluri (1998a, 1998b), KABBANI and PATTY (1992), and TALLURI (1996, 1998). Most of this work is heuristic in nature and does not guarantee that an optimal aircraft routing solution will be determined. Kabbani and Patty (1992) model the aircraft routing problem as a set partitioning model, where the columns represent week-long routings, ignoring maintenance constraints. DE-

SAULNIERS et al. (1994a) present a general modeling and algorithmic framework for routing and scheduling problems but do not provide details of applications or implementations. Clarke et al. (1997) present a flight-based model for the aircraft routing problem and describe a lagrangian relaxation solution approach that adds subtour and maintenance constraints as they are violated.

We show how our string-based model and branch-and-price solution approach for the fleet assignment problem can be used to solve the aircraft routing problem as well.

CONTRIBUTIONS AND OUTLINE OF PAPER

This work makes the following contributions, and is organized as follows:

- In Sections 1 and 3, we present string-based models for fleeting and routing. These models are more flexible than conventional models in that nonlinear, complex costs and constraints can often be modeled easily.
- In Sections 1 and 2, we present a string-based model and branch-and-bound solution approach using column generation for a class of fleet assignment and aircraft routing problems that do not require equal aircraft utilization. Fleeting and routing traditionally have been solved separately. Our fleet assignment model differs from traditional models in that maintenance constraints *and* through revenues are modeled. The novel result is that we simultaneously solve the fleet assignment and aircraft routing problems. Furthermore, we provide proof-of-concept using data provided by an airline. We demonstrate the applicability of the approach and show that unlike existing sequential solution approaches, our simultaneous approach finds feasible solutions when they exist.
- In Section 3, we consider the aircraft routing problem with side constraints requiring equal aircraft utilization. We show how to expand our string-based model to capture these complicating constraints and how to solve this problem with a branch-and-bound solution approach involving column and cut generation. Very little can be found in the literature (one exception is NEMHAUSER and PARK (1991)) on combined cut and column generation. Again, we provide computational results using data provided by an airline and illustrate how equal utilization affects the objective function.

1. A STRING-BASED FLEETING AND ROUTING MODEL

As we stated, a string is a sequence of connected flights that begins and ends at (possibly different) maintenance stations, satisfies flow balance, and is maintenance feasible. An *augmented string* is a string with the minimum time necessary to perform maintenance attached to the end of the last flight in the string. Let S be the set of augmented strings. The cost of a string may include the (negative) through revenues associated with the connecting flights it contains, as well as the operating costs and expected revenues associated with the assigned aircraft type. The cost of an augmented string also includes maintenance costs.

The objective of our string model is to select the set of augmented strings such that each flight segment is assigned to exactly one fleet; and, for any fleet, its assigned flights are partitioned into a set of rotations, or routes beginning and ending at the same location, with each aircraft in the fleet assigned to at most one rotation and total costs minimized. To enforce fleet size limits, we select a single point in time, the *count time*, at which to count aircraft on the ground and in the air. The particular time chosen does not matter since aircraft flow balance is always maintained. For computational reasons, however, we try to select a point in time that minimizes the number of aircraft in the air.

For each fleet $k \in K$, where K is the set of fleets, our model includes two types of variables, *augmented string variables* denoted by x^k and *ground variables* denoted by y^k. An augmented string variable x_s^k for string s is equal to 1 if $s \in S$ is flown by fleet k in the solution, and equal to 0 otherwise. The ground variables y^k are used to count the number of aircraft of fleet k on the ground at maintenance stations. To do this, we define the set of events for k at a station to be the set of flights arriving at or departing from that station. For any flight $i \in F$, where F is the set of flights, its event time for k is its arrival time plus minimum maintenance time if i is an arriving flight and is its departure time otherwise. Then, for each maintenance station, we sort and number all events in increasing order of time. If a tie involving both arriving and departing flights occurs, we give priority (i.e., a lower event number) to the events corresponding to arriving flights, and if the tie involves only arrivals or only departures, we break the ties arbitrarily. We let $e_{i,a}^k$ ($e_{i,d}^k$) be the event number for fleet k corresponding to the arrival (departure) of flight i at some maintenance station, $e_{i,a}^{+,k}$ ($e_{i,d}^{+,k}$) denote the next event for k at that station after the arrival (departure) of i, and $e_{i,a}^{-,k}$ ($e_{i,d}^{+,k}$)

denote the event at that station preceding the arrival (departure) of i. Then, for each flight i arriving at (departing from) a maintenance station, ground variables $y^k_{(e^{-,k}_{i,a}, e^k_{i,a})}$ and $y^k_{(e^k_{i,a}, e^{+,k}_{i,a})}$ ($y^k_{(e^{-,k}_{i,d}, e^k_{i,d})}$ and $y^k_{(e^k_{i,d}, e^{+,k}_{i,d})}$) equal the number of aircraft of fleet k on the ground at that station between the predecessor event of i and the arrival (departure) of i and between the arrival (departure) of i and its successor event, respectively.

We denote the set of ground variables for fleet k as G^k, the set of augmented strings ending with flight i and maintenance as S_i^-, and the set of augmented strings beginning with flight i as S_i^+. We let a_{is} equal 1 if flight $i \in F$ is in augmented string s, and equal 0 otherwise, c^k_s is the cost of flying augmented string s with fleet k, r^k_s is the number of times augmented string s assigned to fleet k crosses the count time, p^k_j is the number of times ground arc $j \in G^k$ for fleet k crosses the count time, and N^k is the number of planes in fleet k. Depending on the relative durations of T and the strings, r^k_s may be any nonnegative integer because a string may be longer than T time units. However, p^k_j will always equal 0 or 1 because ground arcs can be at most T time units long, by definition. Observe that since our flight schedule is repeating for some period T, strings may begin at time t and end at some earlier time $t - \delta$.

Our string-based fleeting and routing model is:

$$\min \sum_{k\in K} \sum_{s\in S} c^k_s x^k_s$$

$$\sum_{k\in K} \sum_{s\in S} a_{is} x^k_s = 1, \quad \forall i \in F$$

$$\sum_{s\in S_i^+} x^k_s - y^k_{(e^{-,k}_{i,d}, e^k_{i,d})} + y^k_{(e^k_{i,d}, e^{+,k}_{i,d})} = 0, \quad \forall i \in F, \quad \forall k \in K$$

$$-\sum_{s\in S_i^-} x^k_s - y^k_{(e^{-,k}_{i,a}, e^k_{i,a})} + y^k_{(e^k_{i,a}, e^{+,k}_{i,a})} = 0, \quad \forall i \in F, \quad \forall k \in K$$

$$\sum_{s\in S} r^k_s x^k_s + \sum_{j\in G^k} p^k_j y^k_j \leqslant N^k, \quad \forall k \in K$$

$$y^k_j \geqslant 0, \quad \forall j \in G^k, \qquad \forall k \in K$$

$$x^k_s \in \{0, 1\}, \quad \forall s \in S, \quad \forall k \in K. \tag{1}$$

The objective is to minimize the total cost of the selected strings. The first set of constraints are cover constraints that ensure each flight is in exactly one string. The second and third sets of constraints, called the *flow balance* constraints, ensure that the number of aircraft of fleet k arriving at and departing from a location are equal. The fourth set of constraints, the count constraints, use the string and ground variables to ensure that the total number of aircraft in the air and on the ground does not exceed the size of fleet k. The last sets of constraints force the number of aircraft on the ground to be non-negative and the number of aircraft assigned to a string to be 0 or 1. We are able to relax the integrality constraints on the y variables because every y is a sum of binary x variables.

Any solution to the string-based fleeting and routing model defines a set of fleet-specific rotations. Although particular connections between strings are not specified, maintenance feasible aircraft routings can always be derived from any solution to the string-based model because with augmented strings, maintenance is performed after the last flight in each string and aircraft balance between strings is satisfied by the flow balance constraints.

For our string-based model, the number of ground variables is limited by the sum of the number of flights terminating or starting at a maintenance station, while the number of strings can be exponential in the number of flights. In solving this model, we include *all* ground variables but only a *subset* of string variables, i.e., a sufficient number to find an optimal solution, as detailed in the next section.

2. STRING-BASED MODEL SOLUTION

We solve our string-based fleeting and routing model using a *branch-and-price* algorithm. Branch-and-price is branch-and-bound with a linear programming (LP) relaxation solved at each node of the branch-and-bound tree using *column generation* (BARNHART et al. (1998)).

2.1 LP Solution

Column generation is used to solve large LPs when the constraint matrix is too large to enumerate explicitly. Starting with a subset of variables, the column generation algorithm determines a set of optimal dual values for this restricted problem. These dual values are used to *price out*; that is, to compute reduced costs of nonbasic variables. In a minimization, a variable with a negative reduced cost may improve the solution and therefore should be added to the restricted problem. Adding columns to the constraint matrix is referred to as *column generation*.

The linear program with a restricted subset of the variables is called the *restricted master problem*. The repeated solution of a restricted master problem followed by the generation of columns continues until no variables have negative reduced cost and optimality is achieved. Column generation works best when negative reduced cost columns can be generated without examining *all* variables. Generating such columns or determining that none exist is called the *pricing subproblem*. The column generation algorithm to solve the LP involves the following steps.

Step 1. Solve the Restricted Master Problem: Find an optimal solution to the current restricted master problem containing only a subset of augmented string variables, and all ground variables.
Step 2. Solve the Pricing Subproblem: Generate columns with negative reduced cost. If no columns are generated, STOP; the LP is solved.
Step 3. Construct a New Restricted Master Problem: Add the columns generated by the pricing subproblem to the restricted master problem; go to Step 1.

While Steps 1 and 3 can be carried out using optimization software such as CPLEX (1993), Step 2 (the pricing subproblem) must be tailored to exploit problem structure.

2.2 Pricing Subproblem

Let π_i be the dual variable associated with the cover constraint for flight i, $\lambda_i^{k,b}$ and $\lambda_i^{k,e}$ be the dual variables corresponding to flow balance constraints for fleet k and augmented strings beginning and ending with flight i, respectively, and σ^k be the (nonpositive) dual variables corresponding to the count constraint for fleet k. Then, the reduced cost of an augmented string s beginning with flight m and ending with flight n is

$$\bar{c}_s^k = c_s^k - \sum_{i \in F} a_{is}\pi_i - \lambda_m^{k,b} + \lambda_n^{k,e} - r_s^k\sigma^k. \qquad (2)$$

The pricing subproblem can be cast as the problem of finding constrained shortest paths over specially constructed networks, described in the next section.

2.3 Flight Networks

The combined fleeting and routing model (1) is derived on a directed *time-line* network, one for each fleet, with the arc set representing the set of flights and the set of ground variables. The nodes in the graph correspond to either the departure or arrival of a flight. The time associated with a flight's departure node is its scheduled departure time, and the time associated with its arrival node is its scheduled arrival time plus minimum turn time. Arcs representing ground variables connect consecutive events (i.e., nodes) at a single location, with a *wrap-around* arc connecting the first and last event at a location. The cost of each ground arc is zero and the cost of each flight arc is the sum of the fleet's operating costs for the flight plus (the negative of) the expected revenues of the flight-fleet assignment.

Through revenues and maintenance costs cannot be captured in the time-line network since they cannot be allocated directly to network arcs. We can, however, capture the maintenance costs in a modified *maintenance augmented time-line* network containing *maintenance arcs*. A maintenance arc is created for each flight j ending at a maintenance station. Like the flight arc for flight j, the tail node of the maintenance arc is the departure node of j. The head node, however, is a *maintenance* node at the arrival maintenance station with associated time equal to the scheduled arrival time of j plus minimum maintenance time. The cost of this maintenance arc is the sum of the associated maintenance costs and the flight arc cost for j.

It is impossible to model through revenues as arc costs in the time-line network because connections between flights are not explicit. Instead, to do this, we use a *connection* network. In a connection network, the node set represents the set of flights and arcs correspond to *connections* between flights. A connection arc exists from one flight i to another flight j if the arrival station of i is also the departure station of j and the arrival time of i plus minimum turn time is earlier than the departure time of j. For stations equipped to carry out maintenance, both connection arcs and *maintenance connection* arcs are included. A maintenance connection arc, representing a *maintenance opportunity*, exists from flight i to another flight j if the arrival station of i is also the departure station of j and j is a maintenance station. The duration of a connection arc from flight i to flight j is the elapsed time between the arrival of i and the departure of j if this time exceeds the minimum maintenance time; and is the elapsed time plus T otherwise.

Depending on the arc's duration, the number of times it crosses the count time is 0 or 1. We build any flight-related costs into the costs on the arcs connecting *from* that flight. In addition, maintenance costs are placed on maintenance connection arcs and (negative) through revenues are placed on connection arcs.

For the time-line and connection networks, an augmented string corresponds to a path beginning and ending with nodes at maintenance stations, with the last arc in the path being a maintenance arc in the time-line network and a maintenance connection arc in the connection network.

Connection networks provide richer modeling possibilities than time-line networks because all costs can be allocated to network arcs. The advantage of a time-line network is that it generally contains many fewer arcs. However, this computational advantage is dominated by the advantages of connection networks, and thus, we use a connection network to implement the pricing routine of Section 2.4 and the branching strategy of Section 2.5. A connection network will also be used for modeling the problem of Section 3.

2.4 Pricing Subproblem Solution

The pricing subproblem is the problem of finding a sequence of flights that satisfy constraints on the number of hours permitted between maintenance checks, and prices out to have a negative reduced cost. The *maximum flying time* requirement restricts the total flying time between maintenance to β hours, while the *maximum elapsed time* requirement restricts the total elapsed time between maintenance to α hours. In either case, we model the pricing subproblem as a resource-constrained shortest path problem in the fleet-specific connection network, described in Section 2.3.

We define C^k to be the set of connection arcs for fleet k, C_M^k to be the set of maintenance connection arcs for fleet k, $F^+ \subset F$ to be the set of flights departing from a maintenance station, and $F^- \subset F$ to be the set of flights arriving at a maintenance station. A simple path from any node in F^+ to any node in F^- in the connection graph *without* maintenance connection arcs represents a string if it is maintenance feasible.

Using the terminology of DESROCHERS (1986), the pricing subproblems can be cast as *resource-constrained shortest path* problems. In the case of the maximum flying time (maximum elapsed time) requirement, each node in a path uses up an amount of *resource* equal to the number of flying hours (total hours) in the flight represented by the node. In the case of the maximum elapsed time requirement, each connection arc in a path also uses up an amount of resource equal to the total elapsed time of the connection represented by the arc. In the maximum flying time (maximum elapsed time) case, a path cannot, in total, consume more than an amount β (α) of resource.

We define the *length* of each arc in the connection network so that the length of a path corresponds to the reduced cost of the flight sequence it represents. To facilitate this definition, we add two dummy nodes f^+ and f^-, neither of which consume any resource, and arcs (f^+, f) for each $f \in F^+$, (f, f^-) for each $f \in F^-$. Let $\bar{F}$ and $\bar{C}^k$ denote the augmented node and arc sets in the connection network for fleet k, i.e.,

$$\bar{F} = F \cup \{f^+, f^-\}$$

and

$$\bar{C}^k = C^k \backslash C_M^k \cup \{(f^+, f): f \in F^+\} \cup \{(f, f^-): f \in F^-\}.$$

Now there is clearly a one-to-one correspondence between maintenance feasible strings and resource-feasible paths from f^+ to f^- in the connection network $(\bar{F}, \bar{C}^k)$ for fleet k. We define the length of any arc (f, g) in $(\bar{F}, \bar{C}^k)$ for each k as follows:

$$l^k(f, g) = \begin{cases} c_f^k + c_{(f,g)}^k - \pi_f - r_f^k\sigma^k - r_{(f,g)}^k\sigma^k, & (f, g) \in C^k \backslash C_M^k \\ c_{f^m}^k - \pi_f - r_{f^m}^k\sigma^k + \lambda_{f^m}^{k,e} & g = f^- \\ -\lambda_g^{k,b} & f = f^+ \end{cases}$$

where c_f^k denotes any costs associated with flight f and fleet k, $c_{(f,g)}^k$ denotes any costs associated with connection (f, g) for fleet k (for example, negative through-value of the connection), $c_{f^m}^k$ denotes any costs associated with flight f followed by maintenance for fleet k, r_f^k is the minimum number of times flight f assigned to fleet k crosses the count time, $r_{(f,g)}^k$ is the minimum number of times the connection (f, g) assigned to fleet k crosses the count time, $r_{f^m}^k$ is the minimum number of times flight f followed by maintenance assigned to fleet k crosses the count time, and π, σ^k, $\lambda^{k,b}$, and $\lambda^{k,e}$ are the dual variables for fleet k as defined in Section 2.2. Clearly the length of any path $(f^+, f_1, \ldots, f_p, f^-)$ for each k is

$$\begin{aligned} & l^k(f^+, f_1) + \sum_{j=1}^{p-1} l^k(f_j, f_{j+1}) + l^k(f_p, f^-) \\ &= -\lambda_{f_1}^{k,b} + \sum_{j=1}^{p-1}\{c_{f_j}^k + c_{(f_j,f_{j+1})}^k - \pi_{f_j} - r_{f_j}^k\sigma^k \\ &\qquad - r_{(f_j,f_{j+1})}^k\sigma^k\} + c_{f_p^m}^k - \pi_{f_p^m} \\ &= \sum_{j=1}^{p-1} c_{f_j}^k + c_{f_p^m}^k + \sum_{j=1}^{p-1} c_{(f_j,f_{j+1})}^k - \sum_{j=1}^{p} \pi_{f_j} - \lambda_{f_1}^{k,b} + \lambda_{f_p^m}^{k,e} \\ &\quad - \left(\sum_{j=1}^{p-1} r_{f_j}^k + r_{f_p^m}^k + \sum_{j=1}^{p-1} r_{(f_j,f_{j+1})}^k\right)\sigma^k \\ &= c_s^k - \sum_{f \in F} a_{fs}\pi_f - \lambda_{f_1}^{k,b} + \lambda_{f_p^m}^{k,e} - r_s^k\sigma^k \end{aligned}$$

where s is the augmented string $(f_1, \ldots, f_p)$.

This shows that the problem of finding a negative cost augmented string can be cast as the problem of finding a shortest resource-constrained path from f^+ to f^- in the connection network $(\bar{F}, \bar{C}^k)$ for fleet k, with lengths l^k as defined above. If the shortest path has length equal to or greater than zero for each k, then there are no negative reduced cost augmented strings, so the current restricted master problem solution is an optimal LP solution; otherwise augmented strings corresponding to negative length paths can be added to improve the solution.

2.5 IP Solution

An integer solution to the fleeting and routing problem may be obtained using two types of branching rules, one that branches on *flight-fleet pairs* and one that branches on *follow-ons*. Branching on flight-fleet pairs results in a partition of flights into fleets, or equivalently, one aircraft routing problem for each fleet. Then, as detailed below, branching on follow-ons achieves an integer solution for each of the resulting routing problems.

Given a fractional solution to the fleeting and routing problem LP relaxation, we can identify a flight f that is assigned to more than one fleet. Our flight-fleet pair branching forces f to be covered by exactly one fleet, say $\bar{k}$, on the left branch, and disallows this $(f, \bar{k})$ assignment on the right branch. To ensure that the pricing subproblem does not generate strings for $\hat{k}$ containing f (when f is assigned to $\bar{k} \neq \hat{k}$ or f cannot be flown by $\hat{k}$), we eliminate from the connection network for $\hat{k}$, all nodes representing flight f. (Or equivalently, we set $c_f^{\hat{k}}$ to a very large positive number.) Note that this is not done directly on individual string variables, but on many variables that contain the flight f.

Once branching on flight-fleet pairs partitions the flights into fleets, we use the ideas of RYAN and FOSTER (1981) and branch on *follow-ons*. Beginning with a fractional solution x^k for some k, we identify two fractional strings $\bar{s}$ and $\hat{s}$ both containing at least one common flight. There must be such a flight, call it i_1, since each flight is covered exactly once in any solution. Further, because each pair of strings in a solution cannot contain exactly the same set of flights, there must be another flight, call it i_2, that is contained in $\bar{s}$ but not $\hat{s}$ and with the property that i_1 is followed immediately by i_2 in string $\bar{s}$. We define $FO_l^k \subset S$ as the set of strings assigned to fleet k with each string containing flight i_1 followed by i_2, and we let $FO_r^k \subset S$ be the set of strings containing flight i_1 and/or flight i_2 and i_1 is not followed by i_2. To branch on follow-ons, we create a left and a right branch. On the left branch, we force flight i_1 to be followed by i_2; that is, we require that $\Sigma_{s \in FO_l^k} x_s^k = 1$. To ensure that the pricing subproblem generates strings satisfying this rule, we eliminate from the connection network for fleet k all arcs connecting i_1 to any flight other than i_2 and all arcs connecting to i_2 from any flight other than i_1.

On the right branch, we do not allow flight i_1 to be followed by i_2, that is, we require that $\Sigma_{s \in FO_r^k} x_s^k = 1$. To ensure that the pricing subproblem generates only strings satisfying this rule, we eliminate from the digraph all arcs connecting i_1 to i_2.

The validity of this branching approach is proved in Barnhart et al. (1998).

2.6 Proof of Concept

Long-haul problems, with mostly longer, international flights, and short-haul problems, with mostly shorter domestic flights, are further differentiated by differences in schedule frequency and network structure. Short-haul flights typically are flown daily, whereas long-haul flight schedules are often repeated weekly. Moreover, short-haul operations are characterized by *hub-and-spoke* networks, and long-haul operations are often *point-to-point* networks. In short-haul operations, many flights arrive at, or depart from, a hub within periods of high activity called *banks*, each of which is associated with several possible assignments of aircraft, or *tail numbers*, to flights. The point-to-point long-haul networks, on the other hand, are characterized by sparse activity at stations, yielding few possible aircraft assignments.

Because of the sparse activity in long-haul problems, a sequential process of first solving the traditional long-haul fleet assignment problem and then the aircraft routing problem will often *not* allow the generation of routes that satisfy maintenance criteria. One strength of our string-based model is that any resulting fleet assignment and routing is guaranteed to satisfy maintenance criteria.

We applied our string-based fleeting and routing solution approach to data provided by a long-haul airline. The long-haul operation consisted of a weekly schedule (that is, $T = 7$ days) of 1124 flights, visiting 40 cities, with 9 fleet types containing a total of 89 aircraft. All computations were run on an IBM RS-6000/370 using the CPLEX v3.0 callable library (1993), and MINTO v.2.0a (NEMHAUSER, SAVELSBERGH, and SIGISMONDI (1994)).

Our first step was to determine if quality solutions could be obtained using the conventional sequential solution approach that solves the fleet assignment problem with aggregate maintenance constraints and then solves, for each fleet, the resulting aircraft routing problems. We found that for five of the nine fleets, maintenance-feasible aircraft routings could *not* be achieved.

Next, we applied our string-based approach for simultaneous fleeting and routing. We applied the concepts of node consolidation and islands as presented in Hane et al. (1995) with the result that the number of constraints in our string-based model was

TABLE I

Fleeting and Routing Model Size

Description	Value
Formulation	
Flight cover constraints	1124
Total balance constraints	648
Fleet count constraints	9
Total constraints	1781

reduced from 6557 to 1781 (Table I). To manage the number of columns in our restricted master problem, we deleted all variables except the ground variables, and the basic variables whenever the total number of variables exceeded 10,000.

In our test case, the elapsed time between maintenance stops was limited to 4 days, and no limits were placed on the flying time. Because only the elapsed time maintenance requirement had to be enforced, we modified our pricing problem solution so that the resource-constrained shortest path computations could be replaced by simple shortest path computations. To do this, we evaluated the elapsed time between the start of flight i and the arrival of flight j, for each i, j pair with $i \in F^+$ and $j \in F^-$. Then, for each pair of flights whose elapsed time did not exceed the maximum allowed, we found a shortest path in the connection network for fleet k beginning with $i \in F^+$ and ending with $j \in F^-$. Because the maximum elapsed time between maintenance is less than T (e.g., the maximum elapsed time between maintenance is 4 days and $T = 7$ days), our shortest path algorithms never encounter negative cost cycles since all cycles in the connection network have elapsed time of *at least* T days, by definition.

The LP relaxation solution required about 4.5 hours to solve, with over 88,000 columns generated. In the interest of evaluating time-solution quality tradeoffs, we defined a *tolerance value* and generated columns at a node only if the solution value at that node exceeded the root node LP solution value by the selected tolerance. We obtained results for four different tolerance values, ranging from 0.25% to 1.0%. The optimality gap, measuring the difference between the best integer solution found and the LP lower bound expressed as a percentage of the LP lower bound, was 1.52% using a tolerance of 0.25%, and the total solution time was 5 hours 39 minutes. Increasing the tolerance had little effect on solution times. For a tolerance of 1.00%, the solution time was 5 hours 27 minutes.

Because this is a planning model, a 5-hour solution time is acceptable and, most importantly, gives results that satisfy the maintenance requirements.

3. ROUTINGS WITH SPECIAL CONSTRAINTS

In their short-haul operations, many airlines place a high priority on having even wear and tear across the aircraft in a fleet. However, the aircraft flying different segments of the flight schedule will encounter different conditions with respect to flight length, weather, other environmental conditions, and maintenance schedules. In the face of this diversity across the schedule, one method by which airlines can ensure even wear and tear on the aircraft in a fleet over the *long term* is to require that every aircraft fly *all* the flights assigned to its fleet. In this way, each aircraft will experience approximately the same mixture of conditions, ensuring equal utilization and equivalent maintenance histories at the end of the schedule period.

Because of the flexibility inherent in the hub-and-spoke nature of short-haul flying, the sequential approach to short-haul fleeting and routing is much less likely to suffer the difficulties encountered in long-haul planning. Furthermore, the hub-and-spoke structure greatly increases the number of possible permutations of flights that can form a rotation, making the routing problem on its own a very challenging one. So, for the purpose of solving short-haul problems with equal utilization requirements, we consider only the routing aspects of the problem; i.e., we assume that the fleeting has been determined *a priori*. In principle, the approach to routing we present here could be combined with fleeting in the manner described in the previous sections.

Because short-haul rotation problems are usually approached as daily problems, with exceptions handled later, we work with the assumption that each flight is flown every day; i.e., that the schedule is a repeating daily schedule.

3.1 Connectivity Constraints

Our intent is to assign the *same* sequence of flights to every aircraft in the fleet. Because two different aircraft cannot fly the same flight at the same time, the assignment must have the following properties:

1. aircraft *start* their flying at different points in the sequence,
2. the sequence is a cycle,
3. the sequence includes every flight assigned to the fleet,
4. the fleet's daily flying is partitioned among the aircraft in the fleet and the partitions are ordered so that an aircraft can fly each partition in turn, one for each day, and

5. after a number of days equal to the number of operational aircraft in the fleet, every aircraft will have flown every flight assigned to the fleet exactly once.

Connectivity constraints are used to enforce the condition that every aircraft flies a cycle containing all of the flights. The connectivity constraints are imposed on the *connection* network because the time-line network cannot capture the flow of individual aircraft. For the rotation to be a cycle containing all of the flights, it must be that the strings and maintenance connection arcs used in the rotation induce a directed Hamiltonian cycle on the connection network. Hence we can use constraints closely related to the subtour elimination constraints of the Asymmetric Traveling Salesman Problem (ATSP) to enforce connectivity.

We define a connectivity constraint for each proper subset of the nodes. For each set $\hat{F} \subset F$ with $2 \leqslant |\hat{F}| \leqslant \lfloor |F|/2 \rfloor$, we define $\delta^+(\hat{F})$ to be the set of connection arcs outgoing from $\hat{F}$, i.e.,

$$\delta^+(\hat{F}) = \{(i, j) \in C : i \in \hat{F},\ j \in F \backslash \hat{F}\},$$

and require that

$$\sum_{s \in S} \eta_{\hat{F}}(s)\, x_s + \sum_{j \in C_M \cap \delta^+(\hat{F})} z_j \geqslant 1 \quad \forall \hat{F} \tag{3}$$

where $\eta_{\hat{F}}(s) = 1$ if string s leaves the set $\hat{F}$, $\eta_{\hat{F}}(s) = 0$ otherwise, and z_j is a binary variable indicating the use of maintenance connection arc $j \in C_M$. The set of constraints (3), which requires that at least one string or at least one maintenance connection arc must leave every subset of flights, ensures that the rotation is a cycle through the entire set of flights. Although they are similar, these constraints are *not* identical to the usual ATSP subtour elimination constraints since a string may leave a set of flights more than once while an arc can leave a set at most once.

We now present the entire equal utilization routing model. Because this model applies to the connection network rather than the time-line network it is somewhat different to that presented in (1), even apart from the connectivity constraints. In addition to dropping the fleet superscript, we define M_i^+ to be the set of maintenance connection arcs having outgoing flight i and M_i^- to be the set of maintenance connection arcs having incoming flight i. Now the revised model is:

$$\min \sum_{s \in S} c_s x_s \tag{4}$$

such that

$$\sum_{s \in S} a_{is} x_s = 1, \quad \forall i \in F$$

$$\sum_{s \in S_i^+} x_s - \sum_{j \in M_i^+} z_j = 0, \quad \forall i \in F$$

$$\sum_{s \in S_i^-} x_s - \sum_{j \in M_i^-} z_j = 0, \quad \forall i \in F$$

$$\sum_{s \in S} r_s x_s + \sum_{j \in M} p_j z_j \leqslant N,$$

$$\sum_{s \in S} \eta_{\hat{F}}(s)\, x_s + \sum_{j \in C_M \cap \delta^+(\hat{F})} z_j \geqslant 1,$$

$$\forall \hat{F} \subset F, \quad 2 \leqslant |\hat{F}| \leqslant \left\lfloor \frac{|F|}{2} \right\rfloor,$$

$$z_j \in \{0, 1\}, \quad \forall j \in C_M$$

$$x_s \in \{0, 1\}, \quad \forall s \in S$$

Because there are a very large number of connectivity constraints, we do not attempt to determine them *a priori*. Instead, we add them only when they are violated by solutions to the LP relaxation. Our approach to finding violated connectivity constraints is described in more detail in the next section.

3.2 Solution Algorithm

Because there are potentially a huge number of columns, and a large number of connectivity constraints, we need to perform both column generation and constraint generation to solve the LP-relaxation of (4) at each node of the branch-and-bound tree.

In solving the LP relaxation at each node of the tree, we alternate between column generation and constraint generation phases. We begin with column generation, which closely follows the description in Section 2.2 as applied to a single fleet. One difference, which we describe in more detail below, arises from the connectivity constraints. Another difference is that the short-haul problem is a daily problem, which may lead to the occurrence of cycles in the shortest path network. Because of the possibility of negative length cycles in this network, we cannot guarantee that a shortest path has been found, and so cannot be sure that the final LP solution is optimal. Once column generation is completed, i.e., no more negative reduced cost columns can be found, we search for connectivity constraints violated by the current LP solution. The problem of finding a violated constraint or showing that none exists is known as the *separation* problem. If a violated connectivity constraint is found, we add it to the LP, and re-start column generation. The cycle repeats until no violated connectivity constraints are found.

We now describe our approach to connectivity constraint generation and the effect of connectivity constraints on the solution algorithm described earlier.

We find violated connectivity constraints by using the usual separation method for the subtour elimination constraints in the ATSP; i.e., by solving a directed minimum capacity cut problem with no designated source and sink nodes. We used the public domain code (from G. Skorobohatyj, ftp.zib.de.in/pub/mathprog/mincut/global-dir) based on the method described in HAO and ORLIN (1992) for solving this problem. With this approach, it is possible, but not likely, that violated connectivity constraints will go undetected in *fractional* LP solutions, because connectivity constraints are not identical to ATSP subtour elimination constraints. However they are sufficiently similar for us to be able to guarantee that connectivity constraints *will* be satisfied by *integer* solutions. This is enough to guarantee the correctness of the algorithm.

An added complication with the connectivity constraints is that we are generating them on an as-needed basis in the context of an LP which is being solved using column generation. Because not all variables are present at the time at which we add a constraint violated by the current solution, the LP with the current set of columns is likely to be infeasible after the constraint has been added. We take care that a feasible solution, albeit artificial, is always available by including an artificial variable in each connectivity constraint, as in a Phase I LP.

The pricing subproblem is also complicated by the presence of the connectivity constraints. Dual variables for these constraints must be included in the arc lengths on the shortest path network in order to correctly price candidate strings.

Suppose that the connectivity constraints for the flight sets $F_1, F_2, \ldots, F_N$ have been added and the corresponding dual multipliers are ω_i for $i = 1, 2, \ldots, N$. Then the reduced cost of a string $s \in S$ beginning with flight k and ending with flight l is

$$\bar{c}_s = \overbrace{c_s - \sum_{i \in F} \pi_i a_{is}}^{a_s} \overbrace{- \lambda_k^b - \lambda_l^e - r_s \sigma}^{\mu_s} - \sum_{i=1}^{N} \eta_{F_i}(s) \omega_i. \tag{5}$$

If the string leaves any of the sets F_i, the corresponding (nonnegative) dual ω_i is subtracted from the reduced cost. Unfortunately, it is impossible to accommodate this modification to the pricing problem purely by modifying the arc lengths in the shortest path network. The reason is that for any set of flights F_i it is possible for a string to leave the set, then return to the set, and leave *again*, i.e., it is possible for a string to use more than one arc in $\delta^+(F_i)$, yet in all cases, regardless of whether the string uses one, two or more arcs in $\delta^+(F_i)$, only *one* multiple of ω_i should be subtracted from the reduced cost.

In the context of a multilabel algorithm for solving the resource-constrained shortest path problem, such as is described in Desrochers (1986) and DESROSIERS et al. (1995), one approach to solving this new pricing problem would be to introduce a new binary resource for each connectivity constraint to indicate whether or not the path had left the corresponding flight set. Arc lengths would then depend on whether this resource had been consumed. However, this approach would increase the computational burden on the pricing algorithm, which is already a bottleneck in the solution procedure, so we adopted the following approximation.

Instead of requiring that the arc lengths yield string lengths which are exactly the reduced costs, we use an arc length which yields an approximation to the reduced cost. We redefine the arc length to be

$$l(f, g) = c_f + c_{(f,g)} - \pi_f - r_f \sigma - r_{(f,g)} \sigma - \sum_{i:(f,g) \in \delta^+(F_i)} \omega_i$$

which results in a string length of $\tilde{c}_s$, according to

$$\tilde{c}_s = \hat{c}_s + \mu_s - \sum_{i=1}^{N} \tilde{\eta}_{F_i}(s) \omega_i$$

where $\tilde{\eta}_{F_i}(s)$ is the *number* of times string s leaves F_i, for each $i = 1, \ldots, N$. Recall that ω_i is nonnegative and $\eta_{F_i}(s)$ is one if s leaves F_i one or more times and zero otherwise. Thus string length $\tilde{c}_s$ may be less than the string's true reduced cost $\bar{c}_s$. So it would be possible to obtain a string s with $\tilde{c}_s < 0$ and $\bar{c}_s \geqslant 0$, in which case we may already have the string or don't need it. In practice, we very rarely found strings with $\tilde{c}_s < \bar{c}_s$, and concluded that for practical purposes, the approximation was a good one.

3.3 Proof of Concept

We tested our branch-and-price-and-cut algorithm on several short-haul routing problems that represent different fleets and that vary in size and availability of maintenance opportunities. All computations were run on an IBM RS-6000/590 using the CPLEX (1993) v4.0 callable library, and MINTO v.2.2 (Nemhauser, Savelsbergh, and Sigismondi (1994)).

For some airlines, the characteristics of the flight schedule vary significantly by fleet. The smaller air-

TABLE II

Short-Haul Data Characteristics

Name	Problem Characteristics: Flights	Connections	Strings	Length
P1	59	264	5,597	6.7
P2	61	299	9,731	8.2
P3	83	877	75,115	7.9
P4	51	414	113,413	9.0
P5	81	1,190	500,873	8.9
P6	105	1,822	1,067,761	9.0
P7	56	429	1,097,603	12.4
P8	58	465	13,357,451	11.6
P9	141	3,390	62,376,469	12.7
P10	190	6,244	⩾500 million	14.2

craft may travel many short out and back flights in a day. The larger aircraft may do one or two long flights.

Table II presents the characteristics of 10 different rotation problems. The column *Flights* indicates the number of flights that are in the rotation on a daily basis. The column *Connections* is the number of possible connections between pairs of flights; i.e., the number of arcs in the connection network. The column *Strings* is the cardinality of the set of all strings that are maintenance feasible. We exhaustively enumerated (with a limit of 500 million) all such strings to get a feel for the size of the problems, but we did not solve these problems using the complete set of strings. The column *Length* is the average length of the strings that were enumerated.

We established several parameters to tune the performance of our algorithm. We initialized our problem with one artificial variable for each flight. These artificial variables appear in the partition and connectivity constraints. We also seeded the problem with several (50–800) randomly derived feasible strings.

We used a "depth-first–best-bound–depth-first" node choice rule. What this rule does is to operate with a depth-first strategy until a feasible solution is found. Then it chooses the next node to be that with the best bound. Then it continues from this node, again using depth-first search, until a feasible solution is found. This cycle is repeated. We set a limit of 1000 nodes evaluated.

For these problems we used the maximum flying time maintenance criterion. This required that the subproblems be constrained shortest path problems.

We present the solutions to the 10 sample short-

TABLE III

Short-Haul Rotation Results

Name	Problem Solutions: Through Value	Strings	Shortest Paths	Cuts	Nodes	Time
P1n	5,350	331	23	0	2	2
P1c	5,252	261	24	3	4	2
P2n	4,688	294	14	0	2	2
P2c	4,590	309	22	1	2	2
P3n	20,204	590	18	0	1	5
P3c	19,472	605	25	4	2	5
P4n	1,247	1,469	2,243	0	1,000	237
P4c	1,247	825	63	7	12	10
P5n	11,486	923	15	0	1	374
P5c	11,486	780	25	5	6	369
P6n	8,219	1,317	21	0	4	457
P6c	8,219	1,583	42	13	9	610
P7n	706	790	52	0	4	23
P7c	Inf	572	16	1	1	25
P8n	1,161	509	28	0	1	8
P8c	Inf	562	35	1	1	9
P9n	25,159	1,816	31	0	17	1,596
P9c	25,159	3,324	60	10	20	3,688
P10n	29,296	3,866	38	0	6	13,352
P10c	29,296	7,357	112	15	21	35,768

haul rotation problems in Table III. In the column *Name* the letter *n* represents problems solved with no connectivity constraints and the letter *c* indicates problems solved with connectivity constraints. *Strings* is the total number of strings generated. The number of shortest path problems solved is in column *Shortest Paths*. The number of connectivity constraints generated is *Cuts*. The number of nodes evaluated and the total time (in seconds) used is in *Nodes* and *Time*, respectively.

In all problems a very small number of strings were generated compared to the total possible. In half the problems, less than 30% of the total solution time was spent generating strings. However, in solving problems *P5*, *P6*, *P9*, and *P10* at least 90% of the total solution time was spent in string generation. Thus improvements in the constrained shortest path method would sharply impact overall solution times for these problems. We plan to address this issue in future research.

Note that all problems except *P4n* solved with very little branching. This shows that the string-based model is very effective. Problem *P4n* is considerably more difficult than problem *P4c*. One might expect the reverse; that the ATSP-like connectivity constraints would make the problem more difficult. We stopped the algorithm after 1000 nodes and although a feasible solution having value 1247 was found, this was not proven to be the best possible. The final upper bound was 1695.

It is interesting to compare the solutions with and without connectivity constraints. Problems *P7* and *P8* are feasible without the connectivity constraints but infeasible with them. This might suggest that at least in some cases, combined fleeting and routing could be useful in short-haul as well as long-haul planning. The objective sometimes, but by no means always, decreases with the addition of connectivity constraints. In five of the eight feasible cases the objective did not decrease. In the remaining 3 cases the largest decrease was only 3.5%. This decrease is a tradeoff that the planners would have to compare to the benefit of equal utilization.

Over all problems, solution times were modest. The longest time required to solve a problem without connectivity constraints was 3.7 hours and, the longest time with connectivity constraints was under 10 hours. For long-term planning problems, these times are reasonable.

REFERENCES

ABARA, J., "Applying Integer Programming to the Fleet Assignment Problem," *Interfaces* **19,** 20–28 (1989).

BALL, P., K. HOFFMAN, AND R. RUSHMEIR, "The Use of Column Generation in Solving Very Large Fleet Assignment Problems," paper presented at *INFORMS Conference*, Washington, D.C., 1996.

BARNHART, C., E. L. JOHNSON, G. L. NEMHAUSER, M. W. P. SAVELSBERGH, AND P. H. VANCE, "Branch-and-Price: Column Generation for Solving Huge Integer Programs," *Oper. Res.*, in press.

CLARKE, L. W., C. A. HANE, E. L. JOHNSON, AND G. L. NEMHAUSER, "Maintenance and Crew Considerations in Fleet Assignment," *Transp. Sci.* **30,** 249–260 (1996).

CLARKE, L. W., E. L. JOHNSON, G. L. NEMHAUSER, AND Z. ZHU, "The Aircraft Rotation Problem," *Ann. Oper. Res. Math. Ind. Syst. II* **69,** 33–46 (1997).

CPLEX OPTIMIZATION INC. *Using the CPLEX Callable Library and CPLEX Mixed Integer Library*, CPLEX Optimization, Incline Village Nevada Version 3, 1993.

DESAULNIERS, G., J. DESROSIERS, I. IOACHIM, M. SOLOMON, AND F. SOUMIS, "A Unified Framework for Deterministic Time Constrained Vehicle Routing and Crew Scheduling Problems," Cahiers du Gerad series, Universite de Montreal, Publication no. G-94-46, 1994a.

DESAULNIERS, G., J. DESROSIERS, M. M. SOLOMON, AND F. SOUMIS, "Daily Aircraft Routing and Scheduling," Technical Report, GERAD, 1994b.

DESROCHERS, M., "An Algorithm for the Shortest Path Problem with Resource Constraints," Centre de Recherche sur les Transports, Universite de Montreal, Publication no. 421A, 1986.

DESROSIERS, J., Y. DUMAS, M. M. SOLOMON, AND F. SOUMIS, "Time Constrained Routing and Scheduling," in *Network Routing*, Handbooks in Operations Research and Management Science, Volume 8, M. O. Ball, T. L. Magnanti, C. L. Monma, and G. L. Nemhauser (eds.) North-Holland, pp. 35–139, 1995.

FEO, T. A., AND J. F. BARD, "Flight Scheduling and Maintenance Base Planning," *Manage. Sci.* **35,** 1415–1432 (1989).

GOPALAN, R. AND K. TALLURI, "Mathematical Models in Airline Schedule Planning: A Survey," *Ann. Oper. Res. Math. Ind. Syst.*, in press.

GOPALAN, R. AND K. TALLURI, "The Aircraft Maintenance Routing Problem," *Oper. Res.*, in press.

HANE, C. A., C. BARNHART, E. L. JOHNSON, R. E. MARSTEN, G. L. NEMHAUSER, AND G. SIGISMONDI, "The Fleet Assignment Problem: Solving a Large-Scale Integer Program," *Math. Programming* **70,** 211–232 (1995).

HAO, J. AND J. B. ORLIN, "A Faster Algorithm for Finding the Minimum Cut in a Graph," *Proceedings of the 3rd Annual ACM-SIAM Symposium on Discrete Algorithms*, Orlando, FL, SIAM, Philadelphia, PA, 165–174, 1992.

KABBANI, N. M. AND B. W. PATTY, "Aircraft Routing at American Airlines," *Proceedings of the Thirty-Second Annual Symposium of the Airlines Group of the International Federation of Operational Societies*, Budapest, Hungary, 1992.

NEMHAUSER, G. L., M. W. P. SAVELSBERGH, AND G. C. SIGISMONDI, "MINTO, a Mixed INTeger Optimizer," *Oper. Res. Lett.* **15,** 47–58 (1994).

NEMHAUSER, G. L. AND S. PARK, "A Polyhedral Approach to Edge Coloring," *Oper. Res. Lett.* **10,** 315–322 (1991).

RYAN, D. M. AND B. A. FOSTER, "An Integer Programming Approach to Scheduling," in *Computer Scheduling of Public Transport: Urban Passenger Vehicle and Crew Scheduling*, A. Wren (ed.), North Holland, Amsterdam, 269–280, 1981.

TALLURI, K., "Swapping Applications in a Daily Airline Fleet Assignment," *Transp. Sci.* **30,** 237–248 (1996).

TALLURI, K., "The Four-Day Aircraft Maintenance Routing Problem," *Transp. Sci.*, in press.

(Received: January 1997; revisions received: October 1997; accepted: October 1997)