

MIT LIBRARIES



3 9080 00570432 2

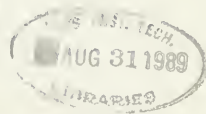






HD28  
.M414  
no.3047-  
89

DEWE



WORKING PAPER  
ALFRED P. SLOAN SCHOOL OF MANAGEMENT

"Selecting Aircraft Routes for  
Long-haul Operations:  
A Formulation and Solution Method"

Anataram Balakrishnan  
T. William Chien  
Richard T. Wong

MIT Sloan School Working Paper #3047-89-MS

July 1989

MASSACHUSETTS  
INSTITUTE OF TECHNOLOGY  
50 MEMORIAL DRIVE  
CAMBRIDGE, MASSACHUSETTS 02139



"Selecting Aircraft Routes for  
Long-haul Operations:  
A Formulation and Solution Method"

Anataram Balakrishnan  
T. William Chien  
Richard T. Wong

MIT Sloan School Working Paper #3047-89-MS

July 1989

MIT, IDADIES

AUG 31 1989

SECRET



Selecting Aircraft Routes for Long-haul Operations:  
A Formulation and Solution Method

Anantaram Balakrishnan

Sloan School of Management  
Massachusetts Institute of Technology  
Cambridge, MA 02139

T. William Chien

Baruch College  
The City University of New York  
New York, NY 10010

Richard T. Wong\*

Krannert Graduate School of Management  
Purdue University  
West Lafayette, IN 47907

\* Supported in part by the Office of Naval Research under University Research Initiative grant # N00014-86-K-0689.



## ABSTRACT

In this paper we consider the routing of long-haul aircraft from a main base to one or more terminal bases. For these long-haul markets, the routing decision becomes critical because route profitability must be evaluated for the extremely large number of feasible routes covered by the operation. In addition, the route selection task is complicated by the "pickup-and-delivery" characteristic of the problem. Therefore, the development of an efficient procedure for selecting good candidate routes will facilitate the iterative flight scheduling process and may lead to more profitable timetables. We define an aircraft routing problem that captures the important profit-generating factors (such as intercity traffic estimates, revenues, operating costs and aircraft capacities) in the route selection decision. We formulate this problem as a mixed integer program, and develop a Lagrangian-based solution procedure that exploits the special structure of the problem. Computational results for several test problems indicate that the procedure is able to select a small number of profitable candidate routes, and provide good bounds that confirm the near-optimality of the generated solutions.



## 1. INTRODUCTION AND LITERATURE REVIEW

The aircraft routing decision is one of the most important components in the overall flight scheduling process of developing a profitable operational timetable of flights for an airline. The flight scheduling process consists of two phases: a schedule construction phase and a schedule evaluation phase. In the schedule construction phase, a set of aircraft routes and the frequency of service on each route are first determined to maximize the profit generated from the operation, while taking into account the traffic estimates and revenue for every origin-destination pair, aircraft characteristics and operating costs, and some operating restrictions. The construction phase is completed by scheduling departure times and assigning aircraft to match the routing and frequency decisions. The resulting timetable is then examined by operating personnel for feasibility and other cost and performance considerations in the schedule evaluation phase. Any desired improvements are then fed back into the construction phase, and a revised set of routes and the associated frequencies are determined. The flight scheduling process iterates between these two phases until a satisfactory final timetable is obtained (Etschmaier and Mathaisel (1984)).

Early research on the aircraft routing problem has focused on applications where the number of alternative routes is relatively small, and mainly used linear programming as the solution method (Dantzig (1963), Kushige (1963), and Miller (1967)). These models represent frequencies as continuous variables; consequently, the solutions are very likely to contain fractional frequencies, which may result in suboptimal integer solutions even after applying some sophisticated rounding procedures. More

recently, researchers have started using mixed integer programming formulations to model and solve the aircraft routing problem. For instance, de Lamotte and Mathaisel (1983) used MPSX-MIP to solve problems of moderate size faced by a small short-haul airline.

However, the literature is still limited in terms of solution methods for larger-scale aircraft routing problems faced by long-haul carriers. For these airlines, the routing decision becomes complex because the large number of intermediate cities covered by their operations dramatically increases the number of feasible routes. Walker-Powell (1970) discusses the Port Linkage Problem for Qantas Airline's Kangaroo Route, which covers 26 cities from Sydney to London. The cities are indexed so that connections are only possible in the direction of increasing order. The airline also imposed additional constraints based on operational considerations. For instance, to simplify the crew scheduling task, several sets of cities, called "slip points", are selected, and every aircraft is required to stop at exactly one city in each set; flights may also terminate at one of the slip points. Between two slip point sets, at most two intermediate stops are permitted for each flight. Given the intercity demand, unit revenue per passenger carried, fixed operating cost for each origin-destination pair, and aircraft capacity restrictions, the Port Linkage Problem seeks a profit maximizing set of routes from Sydney to London (and/or to some slip points). Etschmaier and Richardson (1973) and Richardson (1976) proposed a mixed integer programming formulation for this problem, and developed a Benders' decomposition algorithm to solve it. They reported computational results for problems with up to 17 cities, 2 slip point sets, and at most 9 stops.

The multi-stop aircraft routing problem is related to the pickup-and-delivery routing problem because we must consider both embarkation and disembarkation of passengers at each intermediate stop. The pickup-and-delivery routing problem also arises in the context of school bus routing (Bodin, Golden, Assad and Ball (1983)) and tractor-trailer routing with mixed loads (Ball, Golden, Assad and Bodin (1983)). However, this problem, which is considered more difficult than the pickup-only (or delivery-only) problem, is still largely unexplored by researchers (Golden and Assad (1986)).

Since routing aircraft is the central issue in the schedule construction phase, developing more efficient and effective route selection procedures should result in more profitable timetables. In this paper we address the multiple (homogeneous) aircraft routing problem faced by long-haul carriers. Given traffic estimates and revenues for each origin-destination pair, aircraft operating costs, and aircraft capacities, the aircraft routing problem seeks a set of good candidate routes from a main base to one or more terminal bases in order to maximize total profit. We assume that, for long-haul operations, the cities can be indexed so that connections are only possible from lower indexed nodes to higher indexed nodes. This restriction on the direction of connections is considered practical and even desirable because long-haul markets usually cover very vast geographic areas and the distances between intermediate cities are relatively long. We note here that the aircraft routing problem requires joint consideration of demand selection (i.e., determining the proportion of traffic demand to serve between each origin-destination pair) and routing of aircraft in order to achieve the profit maximization objective.

In this context, the demand selection decision arises because of limited aircraft capacity and differences in the profitability of different origin-destination pairs.

The rest of this paper is organized as follows: We first present a formal definition of the long-haul aircraft routing problem, and formulate it as a mixed integer program in Section 2. Rather than attempting to solve the problem optimally, we focus on devising efficient optimization-based procedures to select good candidate routes and obtain tight upper bounds to evaluate the quality of these results. We use Lagrangian relaxation for this purpose. Section 3 describes our Lagrangian relaxation scheme for the mixed integer programming formulation. We develop an efficient procedure that exploits the special problem structure to solve the Lagrangian subproblems (and obtain upper bounds), and discuss a Lagrangian-based heuristic which constructs good candidate routes (lower bounds). In Section 4 we present computational results for several randomly generated problems with up to 26 cities for the single-aircraft case, and 23 cities for the 4-aircraft case. Finally, in Section 5, we suggest possible extensions of the model.

## 2. PROBLEM DEFINITION AND FORMULATION

The aircraft routing problem that we consider involves routing a fleet of aircraft of the same type (i.e., aircraft having the same capacity and operating costs) from a main base, through several intermediate cities, to one or more terminal bases. As indicated earlier, the cities (including the bases) are indexed so that connections are possible from city  $i$  to city  $j$  only if  $i < j$ . The maximum (forecasted) intercity traffic demand (number



of potential passengers) for each origin-destination (O-D) pair  $(h,k)$  (with  $h < k$ ) is known, and is denoted as  $d_{hk}$ . However, the entire traffic demand need not be satisfied. The airline earns a revenue of  $r_{hk}$  for every passenger it transports from origin  $h$  to destination  $k$ . We assume that the total operating cost is separable by route segments; the operating cost of traveling on segment  $(i,j)$ , i.e., directly from city  $i$  to city  $j$ , is fixed, and denoted as  $f_{ij}$ . As we demonstrate later, our model can also accommodate variable operating costs that depend on the traffic volume on each leg. For simplicity, we ignore these variable costs in our model development. The aircraft routing problem consists of (i) determining the number of passengers to transport between each O-D pair (subject to maximum traffic demand constraints), (ii) assigning the selected traffic to each aircraft (subject to aircraft capacity restrictions), and (iii) routing the aircraft. The problem objective is to maximize total profit.

Every airline imposes some additional operating constraints on its aircraft routes; we do not include all such detailed restrictions in our model. Instead, we focus only on those essential factors, such as traffic demand, revenues, operating cost, and aircraft capacity, that most influence the underlying routing decision of a long-haul carrier. However, the analysis of our basic aircraft routing model should provide useful insights and information for the actual route selection decision. First, since the model captures some of the most relevant profit-determining factors, it can serve as a preliminary screening procedure to assess the market profitability of the regions in which the airline wishes to operate. Secondly, the routes selected by the model represent potentially profitable routes that deserve further examination in the iterative flight scheduling

process. Finally, the profit indicated by this model can serve as a benchmark for cost-effect analysis of other operational and service-level considerations.

The aircraft routing problem is defined over a directed network  $G:(N,A)$  consisting of  $n$  nodes in the set  $N$ , and at most  $n(n-1)/2$  directed arcs belonging to  $A$ . Nodes of the network represent the various intermediate cities and bases, and arcs correspond to potential flight segments. We assume, without loss of generality, that the airline specifies a unique terminal base. Otherwise, we connect the last terminal base to the remaining bases using dummy arcs (that are directed towards the last base) with zero cost. The nodes are indexed from 1 to  $n$ , with indices 1 and  $n$  denoting the main base and terminal base, respectively; indices 2 to  $n-1$  denote the intermediate cities. The network contains a directed arc from city  $i$  to city  $j$  only if  $i < j$ , and direct service from  $i$  to  $j$  is possible. With this network representation, an aircraft route is a directed path from node 1 to node  $n$ . We will use  $(i,j)$  to denote the directed arc or flight segment from  $i$  to  $j$ ;  $(h,k)$  denotes the O-D pair with origin city  $h$  and destination city  $k$ , and  $\langle h,k \rangle$  represents the "commodity" (which will be explained later) between O-D pair  $(h,k)$ . The input parameters for the problem are as follows:

$n$  number of cities (including the bases) covered by the operation,

$V$  number of aircraft (of the same type) at the main base,

$b$  aircraft capacity,

$d_{hk}$  maximum (forecasted) traffic demand for O-D pair  $(h,k)$ ,  $1 \leq h < k \leq n$ ,

$r_{hk}$  unit revenue for O-D pair  $(h,k)$ ,  $1 \leq h < k \leq n$ ,

$f_{ij}$  (non-negative) fixed routing cost per aircraft on flight segment  $(i,j)$   
 $\in A$ .

To formulate the aircraft routing problem as a mixed integer program we define three sets of decision variables:

$s_{hk}$  number of passengers transported from origin city  $h$  to destination city  $k$ ,

$y_{ij}$  number of aircraft traveling on flight segment  $(i,j)$ ,

$x_{ij}^{hk}$  number of passengers originating at city  $h$  and destined for city  $k$  who travel on flight segment  $(i,j)$ ,  $1 \leq h \leq i < j \leq k \leq n$ .

The  $s$ -variables represent the demand selection decision that determines which O-D pairs will be served, and how much of the traffic demand for these O-D pairs will be satisfied. The  $y$ -variables (or, aircraft routing variables) form a set of aircraft routes from the main base to the terminal base. Finally, for each O-D pair  $(h,k)$ , we treat passengers originating at city  $h$  and destined for city  $k$  as a separate commodity (denoted as commodity  $\langle h,k \rangle$ ); the  $x_{ij}^{hk}$  variable denotes the amount of commodity  $\langle h,k \rangle$  transported on segment  $(i,j)$ . Since traffic demand is only a forecasted value representing the expected number of passengers for each O-D pair, the  $s$  and  $x$  variables are assumed to be continuous. Using these decision variables, we formulate the following multi-commodity flow-based mixed integer programming formulation [ARP] for the aircraft routing problem:

[ARP]

$$\text{maximize } \sum_{(h,k)} r_{hk} s_{hk} - \sum_{(i,j)} f_{ij} y_{ij} \quad (1)$$

subject to:

$$s_{hk} \leq d_{hk} \quad \text{for all } (h,k), \quad (2)$$

$$\sum_{j=i+1}^k x_{ij}^{hk} - \sum_{j=h}^{i-1} x_{ji}^{hk} = \begin{cases} +s_{hk}, & \text{if } i=h \\ 0, & \text{if } h < i < k \\ -s_{hk}, & \text{if } i=k \end{cases} \quad \text{for all } \langle h,k \rangle, \quad (3)$$

$$\sum_{\langle h,k \rangle} x_{ij}^{hk} \leq b y_{ij} \quad \text{for all } (i,j), \quad (4)$$

$$\sum_{j=2}^n y_{1j} \leq V \quad (5)$$

$$\sum_{j=i+1}^n y_{ij} - \sum_{j=1}^{i-1} y_{ji} = 0 \quad \text{for } 2 \leq i \leq n-1, \quad (6)$$

$$x_{ij}^{hk} \leq d_{hk} y_{ij} \quad \text{for all } \langle h,k \rangle, (i,j), \quad (7)$$

$$x_{ij}^{hk}, s_{ij} \geq 0 \quad \text{for all } \langle h,k \rangle, (i,j), \text{ and} \quad (8)$$

$$y_{ij} \geq 0, \text{ integer} \quad \text{for all } (i,j). \quad (9)$$

Range of indices:  $1 \leq h \leq i < j \leq k \leq n$ .

The objective is to maximize total profit, which is defined in (1) as the total revenue (the first term) minus the routing costs (the second term). Constraint (2) ensures that the amount of traffic served between each O-D pair does not exceed the corresponding demand. Constraints (3)

are the commodity flow conservation equations. Essentially, they set the net flow of commodity  $\langle h,k \rangle$  leaving origin  $h$  and entering destination  $k$  equal to the number of passengers  $s_{hk}$  transported between O-D pair  $(h,k)$ ; at every other node the inflow equals the outflow for this commodity. Constraint (4) states that the total number of passengers transported through any flight segment should not exceed the total capacity of the aircraft traveling on that segment. Constraints (5) and (6) model the aircraft movements. Constraint (5) specifies that at most  $V$  aircraft can leave the main base. Here, we implicitly assume that each aircraft makes at most one trip during the planning horizon. Constraint (6) specifies that the number of aircraft entering each intermediate city must equal the number of aircraft leaving that city. Finally, we impose the forcing constraint (7) which specifies that passengers cannot travel on a segment  $(i,j)$  unless we schedule aircraft on this segment. Observe that constraint (4) also imposes the same restriction. Hence, constraint (7) is redundant in the integer formulation [ARP]. However, they are not redundant when we relax the integrality restriction on the  $y$  variables; thus, including them in the formulation improves the linear programming relaxation bound.

Notice that, without our assumptions about permissible flight directions (from lower to higher indexed nodes only), the routing problem formulation requires additional constraints to prevent subtours. A subtour is a directed cycle that does not include one or both bases; thus, subtours are not feasible routes. However, when we maximize profit for a routing problem that is defined over a general network, the model may select subtours that have positive net revenue. To prevent this phenomenon, we must add explicit subtour elimination constraints; these constraints

increase the formulation size and make the problem more difficult to solve. Our node indexing assumption makes the network acyclic, thus obviating the need for these additional constraints in our aircraft routing model.

The aircraft routing problem can also be formulated in several alternative ways. For instance, Etschmaier and Richardson (1973) formulate the problem using two sets of decision variables: integral routing variables and continuous single-commodity variables that represent the number of passengers traveling on each flight segment. Although this single-commodity formulation is more compact, its linear programming relaxation (LP) bound is very loose. (The LP bound obtained using their formulation for one small test problem resulted in 120% gap between the LP and integer solutions; the LP solution coincided with the integer solution using our formulation.)

The size of the mixed integer program for [ARP] grows rapidly as the number of nodes increases. For instance, for a 20-city complete network (with all possible directed arcs  $(i,j)$ ,  $i < j$ ), the formulation contains 190 integer variables, 7505 continuous variables, and 9425 constraints. The size is more than doubled when we consider a 26-city problem, for which the formulation will contain 325 integer variables, 20800 continuous variables, and 24400 constraints. Mixed integer programs of this size are relatively difficult to solve optimally using branch-and-bound or other enumeration methods. In particular, the imbedded "pickup-and-delivery" characteristic of the problem complicates the task of evaluating the large number of feasible routes (for instance, a 20-city problem has 262,144 feasible routes; the number of feasible routes increases to 16,777,216 for a 26-city problem). Therefore, we focus on developing an efficient

procedure to identify good (possibly suboptimal) demand selection decision and candidate routes using a Lagrangian relaxation scheme which we introduce in the next section.

### 3. THE SOLUTION METHOD

A number of successful applications of the Lagrangian relaxation approach have been reported in the literature. For instance, Lagrangian-based algorithms have been used effectively to solve the traveling salesman problem (Held and Karp (1970)), the facility location problem (Cornuejols, Fisher and Nemhauser (1977)), and the generalized assignment problem (Fisher, Jaikumar and Van Wassenhove (1980)). Geoffrion (1974), Shapiro (1979), and Fisher (1981) discuss the Lagrangian relaxation approach and its numerous applications. The method consists of removing certain "complicating" constraints, and incorporating them in the objective function using Lagrangian multipliers. The purpose of this relaxation is to obtain a subproblem that can be solved efficiently because of its special structure. For a given set of multipliers, the relaxed problem's objective function value serves as an upper bound (for maximization objective functions) on the optimal profit of the original problem.

In our Lagrangian relaxation-based solution procedure, we dualize the commodity flow conservation equations (3) using Lagrangian multipliers  $u_i^{hk}$  for all  $\langle h, k \rangle$ ,  $h \leq i \leq k$ . Intuitively,  $u_i^{hk}$  represents the imputed routing cost to transport each passenger from origin  $h$ , through city  $i$ , to destination  $k$ . The relaxed problem, denoted as [RP], is given below.



[RP]

$$z_{RP} = \text{maximize } \sum_{(h,k)} \tilde{r}_{hk} s_{hk} - \sum_{(i,j)} f_{ij} y_{ij} - \sum_{(i,j) < h,k >} c_{ij}^{hk} x_{ij}^{hk}$$

subject to [2], (4) - [9].

Where,

$$\tilde{r}_{hk} = r_{hk} + u_h^{hk} - u_k^{hk} \quad \text{for all } (h,k),$$

$$c_{ij}^{hk} = u_i^{hk} - u_j^{hk} \quad \text{for all } <h,k>, (i,j).$$

For a given set of Lagrangian multipliers, the relaxed problem [RP] further decomposes into a demand selection subproblem (containing only the  $s$  variables) and a capacity allocation/route selection subproblem (containing only the  $x$  and  $y$  variables). The selected Lagrangian multiplier values determine the subproblem objective function coefficients, namely, the "adjusted" unit revenues  $\tilde{r}_{hk}$  and the imputed intercity transportation costs  $c_{ij}^{hk}$ . Observe that if the application context imposes variable operating costs (in addition to the fixed costs), these costs can be added to the imputed cost  $c_{ij}^{hk}$ . As we show in the next section the demand selection subproblem can be solved by inspection. On the other hand, the capacity allocation/route selection subproblem can be transformed into a minimum-cost flow problem for the multiple-aircraft case or a shortest path problem for the single-aircraft case. Both these problems can be solved efficiently using specialized network flow algorithms. Solving the demand selection and capacity allocation/route selection



subproblems gives an upper bound to [ARP]. To improve the upper bound, we iteratively update the Lagrangian multipliers using the subgradient search method. We also apply a Lagrangian-based heuristic to construct feasible solutions that provide lower bounds to [ARP].

These different components of the Lagrangian relaxation approach are described in greater detail in the following sections: Section 3.1 describes both the demand selection and the capacity allocation/route selection subproblems, and their solution methods. Section 3.2 discusses the subgradient search procedure that we use to update the Lagrangian multipliers. In Section 3.3, we present a Lagrangian-based heuristic that constructs locally optimal solutions to [ARP].

### 3.1. Solving the Lagrangian Subproblems

The demand selection subproblem, abbreviated as [DSS], consists of only the  $s$  variables, and is defined below:

[DSS]

$$\text{maximize } \sum_{(h,k)} \tilde{r}_{hk} s_{hk} \quad (10)$$

subject to:

$$0 \leq s_{hk} \leq d_{hk} \quad \text{for all } (h,k). \quad (11)$$

Observe that [DSS] can be solved by inspecting the sign of  $\tilde{r}_{hk}$ . That is, if  $\tilde{r}_{hk}$  is positive for O-D pair  $(h,k)$ , this O-D pair is "selected", and the entire demand for this O-D pair is transported; otherwise, no passenger

for this O-D pair is transported. Therefore, the optimal solution to the subproblem [DSS] is:

$$\bar{s}_{hk} = \begin{cases} d_{hk}, & \text{if } \bar{r}_{hk} > 0 \\ 0, & \text{if } \bar{r}_{hk} \leq 0, \end{cases}$$

for all O-D pairs (h,k).

The capacity allocation/route selection subproblem, denoted as [CARSS], contains only the x and y variables and is defined as follows:

[CARSS]

$$\text{minimize} \quad \sum_{(i,j)} f_{ij} y_{ij} + \sum_{(i,j) < h,k} c_{ij}^{hk} x_{ij}^{hk} \quad [12]$$

subject to:

$$\sum_{j=2}^n y_{ij} \leq V \quad [13]$$

$$\sum_{j=i+1}^n y_{ij} - \sum_{j=1}^{i-1} y_{ji} = 0 \quad \text{for } 2 \leq i \leq n-1, \quad [14]$$

$$\sum_{<h,k>} x_{ij}^{hk} \leq b y_{ij} \quad \text{for all } (i,j), \quad [15]$$

$$x_{ij}^{hk} \leq d_{hk} y_{ij} \quad \text{for all } <h,k>, (i,j), \quad [16]$$

$$x_{ij}^{hk} \geq 0 \quad \text{for all } <h,k>, (i,j), \text{ and} \quad [17]$$

$$y_{ij} \geq 0, \text{ integer} \quad \text{for all } (i,j). \quad [18]$$

Essentially, [CARSS] seeks to minimize the total routing costs (12) by finding a set of aircraft routes (defined by (13) and (14)), and allocating

the capacity on each segment of the route among the "commodities"  $\langle h,k \rangle$  (restricted by (15) and (16)).

Observe that since [CARSS] does not have commodity flow conservation requirements, the optimal allocation of capacity among the commodities can be determined separately for each arc. Suppose that there are  $m (\leq V)$  aircraft traveling on arc  $(i,j)$ . The optimal capacity allocation, and the minimum cost corresponding to this aircraft flow can be determined by solving the following continuous knapsack problem [CKP $_{ij}^m$ ]:

[CKP $_{ij}^m$ ]:

$$d_{ijm} = \text{minimize } \sum_{\langle h,k \rangle} c_{ij}^{hk} x_{ij}^{hk} + mf_{ij} \quad (19)$$

subject to:

$$\sum_{\langle h,k \rangle} x_{ij}^{hk} \leq mb \quad (20)$$

$$0 \leq x_{ij}^{hk} \leq d_{hk} \quad \text{for all } \langle h,k \rangle. \quad (21)$$

This continuous knapsack problem can be solved efficiently using the following greedy procedure: Sort the commodities in increasing order of  $c_{ij}^{hk}$ . Consider each commodity  $\langle h,k \rangle$  in sequence from this list. If  $c_{ij}^{hk}$  is negative, set  $x_{ij}^{hk}$  equal to the minimum of commodity  $\langle h,k \rangle$ 's traffic demand ( $d_{hk}$ ) and the remaining aircraft capacity. Stop if the aircraft capacity is exhausted or all remaining commodities have non-negative imputed routing cost  $c_{ij}^{hk}$ .

Notice that  $d_{ijm}$ , which denotes the minimum routing "cost" when  $m$  aircraft travel on arc  $(i,j)$ , is convex in  $m$  since the commodities are

selected in non-decreasing order of  $c_{ij}^{hk}$ . (As the number of aircraft  $m$  increases the incremental benefit of selecting additional passengers decreases.) After computing  $d_{ijm}$  for all arcs  $(i,j)$  and  $m = 1, \dots, V$ , we can solve [CARSS] by finding a minimum-cost flow from node 1 to node  $n$ , with the convex cost function  $d_{ijm}$  on each arc. (Recall that, in [CARSS], the flow on each arc represent the number of aircraft assigned to that flight segment.) If the resulting minimum cost is negative, then the directed paths in the optimal minimum-cost flow solution define an optimal set of aircraft routes, with the number of aircraft on each route equal to the flow on the corresponding directed path; otherwise, no route is initiated.

We next demonstrate how to transform the minimum convex cost problem into an ordinary network flow problem over a suitably modified network. Figure 1 gives a small example illustrating this transformation. The original network contains 3 nodes and 3 directed arcs. We set up a modified network for the equivalent minimum-cost flow problem as follows: To model the convex cost function on each original arc  $(i,j)$ , we introduce  $V$  parallel directed arcs from node  $i$  to node  $j$  with costs equal to  $d_{ij1}$  and  $(d_{ijm} - d_{i,j,m-1})$ , respectively, for the first and the  $m^{\text{th}}$  arcs,  $m = 1, \dots, V$ . Thus, the cost on each parallel directed arc from node  $i$  to node  $j$  is the marginal cost of assigning an additional aircraft to segment  $(i,j)$ . Each directed arc carries an upper limit on flow of 1 unit. In addition to the parallel arcs, the modified network also contains a dummy node  $D$ , and two dummy directed arcs  $(1,D)$  and  $(n,D)$ . These two dummy directed arcs have costs equal to 0, and upper limits on flow equal to  $V$ . The minimum-cost flow problem is solved from node 1 to node  $D$ . If the minimum cost is

zero, no route is selected; otherwise, the optimal arc flows define an optimal set of aircraft routes for subproblem [CARSS].

-----  
 insert Fig. 1 about here  
 -----

For the single-aircraft case, since only one cost coefficient,  $d_{ij1}$ , is necessary for each arc  $(i,j)$ , [CARSS] can alternatively be solved by finding a shortest path from node 1 to node  $n$  using  $d_{ij1}$  as the length for arc  $(i,j)$ . Observe that even though arc lengths may be negative, the shortest path problems can be solved easily since the network does not contain any directed cycles (and, hence, any negative cycles). If the length of the shortest path is negative, the shortest path defines an optimal aircraft route; otherwise, no route is initiated.

### 3.2. Generating Upper Bounds for [ARP]

For any given vector  $U$  of Lagrangian multipliers, the objective function value,  $z_{RP}(U)$ , of the Lagrangian problem [RP] provides an upper bound to [ARP]. To find the best upper bound we must solve the following dual problem:

$$[DP] \quad \min_U z_{RP}(U).$$

Our method attempts to find a near-optimal solution to [DP] by using a subgradient search method to update the multipliers  $u_i^{hk}$  and improve  $z_{RP}(U)$ . Since the multipliers  $u_i^{hk}$  serve as imputed variable costs for transporting passengers from origin  $h$ , through city  $i$ , to destination  $k$ , we initialize  $u_i^{hk}$  as follows:

$$u_i^{hk} = \begin{cases} -r_{hk} / 2, & \text{if } i=h \\ (g_{hi} + g_{ik}) / b, & \text{if } h < i < k \\ r_{hk} / 2, & \text{if } i=k, \end{cases} \quad \text{for all } (h,k)$$

where,  $g_{pq}$  is the shortest path distance from node  $p$  to node  $q$  using the fixed costs  $f_{pq}$  as arc lengths. Effectively, the initialization scheme prorates the fixed routing costs to each passenger traveling from an origin city  $h$ , through an intermediate city  $i$ , to a destination city  $k$ , for  $h < i < k$ . At both the origin city  $h$  and the destination city  $k$ , we use half the unit revenue for O-D pair  $(h,k)$ , and treat it as a negative variable cost for transporting one passenger from city  $h$  to city  $k$ .

Since our relaxation scheme dualizes the commodity flow conservation equations (3) in [ARP], the solutions  $\bar{s}_{ij}$  obtained from subproblem [DSS] and  $\bar{x}_{ij}^{hk}$  from [CARSS] may not conserve flow at each node. The excess outflow of commodity  $\langle h,k \rangle$  at each node  $i$ ,  $h \leq i \leq k$ , defines a subgradient direction for changing the Lagrangian multipliers  $u_i^{hk}$ . Our solution procedure iteratively updates the Lagrangian multipliers using a standard subgradient search method (see, for example, Held, Wolfe and Crowder (1974), and Fisher (1981)), solves the revised Lagrangian subproblems, and computes the new subgradients until either the number of iterations exceeds a prespecified limit or the step-size multiplier value falls below a given threshold. We note here that our relaxation scheme does not satisfy the Integrality Property (Geoffrion 1974), and may, therefore, provide tighter bounds than the LP relaxation of [ARP].

### 3.3. A Lagrangian-Based Heuristic for [ARP]

The Lagrangian solutions, although not necessarily feasible to [ARP],

do provide useful information for constructing feasible solutions. Our Lagrangian-based heuristic uses the optimal solution to [CARSS] to construct an initial feasible solution for [ARP]. Then, we apply a local improvement procedure that attempts to better utilize the aircraft capacity. Recall that the non-zero  $y$  solutions obtained from subproblem [CARSS] define either a single route or a set of multiple routes. In the following discussions, we will focus on the case when multiple routes are obtained from [CARSS].

A step-by-step description of the heuristic follows:

Step 1: Separate the set of routes into  $V_0$  single routes and number them. Set  $m=1$ . Recall that the optimal solution to the equivalent minimum-cost flow problem defines a set of aircraft routes, with the number of aircraft on each route equal to the flow on the corresponding directed path. The heuristic first obtains  $V_0$  single routes by tracing through each directed path and identifying the flow on the directed path.

Step 2: Obtain initial supply amounts from the  $\bar{x}$  solutions from [CARSS]. To get a good initial supply amount  $s_{hk}^0$  for each O-D pair  $(h,k)$  that can be served by the set of routes obtained in step 1, we set  $s_{hk}^0$  equal to the minimum of the outflow of  $\langle h,k \rangle$  from node  $h$ , the inflow of  $\langle h,k \rangle$  to node  $k$ , the demand for O-D pair  $(h,k)$ , and the aircraft capacity. That is,

$$s_{hk}^0 = \min \left\{ \sum_{j=h+1}^k \bar{x}_{hj}^{hk}, \sum_{j=h}^{k-1} \bar{x}_{jk}^{hk}, d_{hk}, b \right\}.$$

This supply assignment is preferable to using the  $\bar{s}$  solution obtained from subproblem [DSS], since this subproblem's "all-or-nothing" solution



characteristic provides little information on how much of the maximum traffic demand  $d_{hk}$  to actually transport.

Step 3: For the  $m^{\text{th}}$  route, verify feasibility with respect to aircraft capacity constraints. Let  $(H,K)_m$  denote the set of O-D pairs  $(h,k)$  that can be served by the  $m^{\text{th}}$  route. Although each individual  $s_{hk}^0$  value, for  $(h,k) \in (H,K)_m$ , satisfies the capacity constraint, the sum of the  $s_{hk}^0$  values on some segments of the route may exceed the aircraft capacity. Step 3 checks for this possible situation and, if aircraft capacity is violated, reduces the supply amounts to those O-D pairs that result in the smallest decrease in profit, until the capacity constraint is met on each segment of the route. Let  $\hat{s}_{hk}$  denote the revised supply amount for the O-D pair  $(h,k) \in (H,K)_m$ .

Step 4: Compute possible improvement in profit by reallocating aircraft capacity. For each O-D pair  $(h,k) \in (H,K)_m$  that has unsatisfied demand, the heuristic computes the marginal profit of reallocating the aircraft capacity by increasing the supply amount to this O-D pair, and correspondingly reducing the supply amount to other O-D pairs, if necessary. If profit improves for none of the O-D pairs, go to step 6.

Step 5: Revise  $\hat{s}_{hk}$  by reallocating the aircraft capacity to the most profitable commodity. The most profitable reallocation in step 4 is performed. After revising the corresponding supply amounts, the heuristic returns to step 4 to check for other possible improvements.

Step 6: If  $m=V_0$ , Stop. Otherwise, check profitability of this route, update  $\{s_{hk}^0\}$ , increase  $m$  by 1, and go to step 3. So far we have obtained an improved capacity allocation. However, the entire route may still produce negative profit. In this case, we simply discard this route,



reduce  $y_{ij}$  by the amount of aircraft flow for each  $(i,j)$  on this route, and set  $\hat{s}_{hk}$  to 0 for all  $(h,k) \in (H,K)_m$ . Finally, the initial set of supply amounts  $\{s_{hk}^0\}$  is revised by setting

$$s_{hk}^0 \leftarrow \max \{ s_{hk}^0 - \hat{s}_{hk}, 0 \} \text{ for all } (h,k) \in (H,K)_m.$$

The heuristic returns to step 3 to examine the next route.

In our implementation, the heuristic is applied periodically (once every 10 iterations) to improve the lower bound until the step-size multiplier reaches a prespecified threshold value. Thereafter, the heuristic is applied after every subgradient iteration.

#### 4. COMPUTATIONAL RESULTS

Since actual data is difficult to obtain due to its confidential nature, we tested the solution procedure on the single-aircraft 12-city problem reported in Etschmaier and Richardson (1973), and on several randomly-generated problems. The test problems reported here are not only larger (in terms of number of cities and aircraft) than the previously attempted problems described in the literature, but also more closely reflect the size of the problems that arise in practice. The test problems were categorized into 15 problem classes. For each problem class, we generated 5 different instances using different seeds for the random number generator. Problem classes 1 to 4 are single-aircraft problems with 17, 20, 23, and 26 cities, respectively. Problem classes 5 to 8 are 2-aircraft problems with 17, 20, 23, and 26 cities, respectively. Problem classes 9 to 12 are 3-aircraft problems with 17, 20, 23, and 24 cities, respectively.

Finally, problem classes 13, 14, and 15 are 4-aircraft problems with 17, 20, and 23 cities, respectively.

Given the desired number of cities (including the bases), the test problem generator first randomly locates the required number of nodes on a 100 x 100 grid, and numbers the nodes in increasing order of the Euclidean distance from the origin (0,0). All our test problems are complete, i.e., the networks contain all the possible directed arcs. Demand for each O-D pair was generated from a uniform (0,100) distribution. Finally, the unit revenue for each O-D pair, and the fixed routing cost on each arc were generated as follows:

$$r_{hk} = EU_{hk} + \text{UNIFORM}(0,10), \quad \text{for all } (h,k),$$

$$f_{ij} = 30 * EU_{ij} + \text{UNIFORM}(0,100), \text{ for all } (i,j),$$

where  $EU_{pq}$  is the Euclidean distance between node p and node q. The aircraft capacity, b, was set equal to 100 for all the test problems.

The solution procedure was coded in FORTRAN and implemented on an IBM 3083 computer. For the single aircraft case, we solved the shortest-path subproblem using the algorithm discussed in Lawler (1976) for acyclic networks; for multiple aircraft problems, we use a modified version of the NETFLO code (Kennington and Helgason (1980)) to solve the minimum-cost flow subproblem. Our implementation does not employ any special data structures; with improvements in data organization, the solution procedure could perhaps solve larger problems.

Our implementation of the subgradient procedure initializes the step-size multiplier to 2.0, and halves the multiplier if the Lagrangian objective function value does not improve for 20 consecutive iterations.

The solution procedure terminates when either the number of iterations exceeds 1000, or the step-size multiplier reduces to a value below 0.01. The local improvement heuristic was applied at the first iteration, and once every 10 iterations until the step-size multiplier reached a value of 0.1; thereafter, the heuristic was applied at every iteration. For each test problem we recorded (i) the initial upper bound (obtained by solving the Lagrangian problem with the initial multipliers), (ii) the best upper bound when the procedure terminates, and (iii) the best lower bound among all heuristic solutions. We use the performance measure  $\%GAP = (UB - LB) / LB * 100\%$ , where UB and LB are respectively the best upper and lower bounds, to evaluate the quality of the solutions.

Table 1 summarizes the performance of the solution procedure for each of the 15 problem classes. Notice that the algorithm was able to reduce the gap between the upper and lower bounds from an initial value of 50.93% to a final value of 1.65% on average, and the largest final %GAP was only 3.22% among the 75 test problems. These statistics show the effectiveness of the subgradient method for improving the upper bounds. We also observed the following interesting characteristic of the heuristic solutions generated by the method. For every problem instance that we tested, subproblem [CARSS] generated only a few alternative routes for step-size multiplier values below 0.1, and the best feasible solution to each problem instance always resulted from one of the routes in this small set. Hence, the solution procedure seems to select a small number of good candidate routes from the fairly large number of possible routes in the problem.

-----  
 insert Table 1 about here  
 -----

The figures for CPU times in Table 1 show that computational requirements grow as the number of nodes and/or the number of aircraft increases. However, the computational requirement increases at a slower rate than the problem size. For instance, the size of a 26-city complete network problem is more than twice the size of a 20-city complete network problem in terms of decision variables and constraints in the problem formulation. Yet, on average the computational requirement was only 1.88 times for the 1-aircraft case and 1.75 times for the 2-aircraft case. Also, increasing the number of aircraft seems to increase the CPU time only marginally. For instance, the computational requirement increases only about 38% on average when the number of aircraft increases from 2 to 4.

Table 2 records the breakdown of total CPU time among the 3 submodules of the solution procedure. For instance, solving the shortest path subproblem accounted for about 1/5 to 1/4 of the total CPU time for the single-aircraft problems; solving the minimum-cost flow subproblem accounted for about 1/3 of the total CPU time for the multiple-aircraft cases. Finally, constructing feasible solutions requires an increasing proportion of the total CPU time as the number of nodes and/or the number of aircraft increases. We note here that the modified NETFLO code constructs, in every iteration, a fresh network and an initial feasible solution for subproblem [CARSS]. In retrospect, since the underlying network for the minimum-cost flow problem remains the same in each iteration, some computational time may be saved if we had modified the NETFLO code to retain the initial network. Furthermore, since [CARSS] generates only limited number of alternative routes (for step-size multiplier values below 0.1), we could use the solution from one iteration

to initialize the next, thus saving some additional computational time.

-----  
 insert Table 2 about here  
 -----

We also tested our solution procedure on the 12-city problem listed in Etschmaier and Richardson (1973). To ensure that our solution will contain exactly one city in the slip point set (cities 7 and 8), we assigned a large fixed routing cost on arcs that skip these nodes (i.e., arcs  $(i,j)$ , with  $1 \leq i \leq 6$  and  $9 \leq j \leq 12$ ), and on arcs interconnecting the nodes of the set (i.e., arc  $(7,8)$ ). The large routing costs on these arcs make direct flights between the preceding (1 to 6) and following cities (9 to 12) unprofitable. Therefore, in order to serve customers originating in cities 1 to 6 and destined for cities 9 to 12, aircraft must stop at either city 7 or city 8. If an aircraft stops at city 7, the large routing cost between city 7 and city 8 prevents a subsequent stop at city 8. Therefore, our solution will contain exactly one city in the slip point set (cities 7 and 8). However, we relaxed Etschmaier and Richardson's original restriction that aircraft can stop at most twice before and after the slip point stop. Table 3 compares our solution with Etschmaier and Richardson's schedule.

-----  
 insert Table 3 about here  
 -----

Observe that by relaxing the maximum number of stops restriction, we obtained a solution with \$1078 higher profit. This value is essentially an opportunity cost for imposing the upper limit on number of stops. Notice

also that the optimal route for the original problem can be derived from our solution by applying a drop heuristic, which deletes one of the first three cities that least decreases the profit. The result of this test problem illustrates the flexibility and usefulness of our model in selecting good candidate routes which must satisfy some specific operating constraints. Similar procedures can be employed to account for other unique operational considerations in constructing aircraft routes for a specific airline.

## 5. SUMMARY

Flight scheduling is one of the most important operational decisions for an airline. The final timetable of flights is the result of several iterations of the route construction and route evaluation phases. For long-haul carriers, the aircraft routing problem in the route construction phase is very complex because selecting a set of good candidate routes requires evaluating a combinatorially large number of feasible routes, and incorporating the "pickup-and-delivery" characteristic of the problem. Therefore, the development of an efficient procedure for selecting good candidate routes will facilitate the iterative flight scheduling process and eventually lead to a more profitable timetable.

In this paper we have discussed a version of the aircraft routing problem that captures some of the most important profit-determining factors in the route selection decision faced by a long-haul carrier that operates in a base-to-base environment. Our model makes the common assumption that cities are visited in increasing order of node indices. We presented a mixed integer programming formulation for the multiple (homogeneous) aircraft problem, and developed a Lagrangian relaxation-based solution



procedure that exploits the structure of the problem. Computational results show that this procedure generates aircraft routes and traffic allocations that are within 1-3% of optimality for several relatively large, random test problems. The proposed analysis of the aircraft routing problem serves as a preliminary screening procedure to assess the market profitability of different regions, identify potentially profitable routes, and facilitate cost-effect analysis of other operational and service level policies.

In addition to the basic problem setting described in this paper, this model can accommodate some operating constraints that may arise in practice. For instance, in the previous section we have illustrated a method to modify the cost matrix in order to ensure that each route serves exactly one city in the slip point set. To comply with other operating constraints such as fixed number of stops between slip point sets, we can apply a drop/add heuristic to the best solution for the basic model. We note that although the Lagrangian upper bounds are still valid for the more restricted problem, we expect the %GAP to increase because imposing additional operating constraints will usually decrease the profit, and hence, the lower bound. As we noted earlier, our model can easily accommodate variable operating costs (that depend on traffic volume) in addition to the fixed cost per aircraft on each flight segment. Application contexts where more than one terminal base exists can be solved using a suitable network representation. Round-trip operations can also be modeled by replicating each node, i.e., adding a mirror image of the original one-way network. The solution procedure can then be applied

to the expanded network with modified demand, revenue, and operating cost matrices.

Several extensions and variants of our basic model merit further investigation. First, while our model can accommodate fixed as variable operating costs, all our test problems assume only a fixed cost for each flight segment. Additional test results for problems with variable costs might be useful. Empirical testing of alternative heuristic methods and multiplier adjustment schemes is also worth pursuing. Second, our model assumes that all aircraft are of the same type. This assumption enables us to use aggregate routing variables (the  $y$  variables) in formulation [ARP], instead of detailed binary variables to trace the route for each individual aircraft. Modeling heterogeneous aircraft fleet (with different operating costs and capacities), on the other hand, will require a detailed formulation that destroys some of the special structure in our model. Our solution algorithm might possibly extend to this more complex model as well. We expect, however, that because of the increased complexity the method's computational requirements will increase and its performance (measured in terms of %GAPs) will deteriorate for problems with multiple aircraft types. Finally, relaxing the node indexing assumption (i.e., the assumption that flights are permitted only in the direction of increasing node indices) might increase the model's applicability to contexts (such as short-haul routes) where the cities are not completely ordered. Again this extension increases the model complexity greatly, and may require different solution approaches. Chien (1987) explores some of these assumptions.



## ACKNOWLEDGEMENTS

The authors wish to thank the two anonymous referees and the Associate Editor for their helpful comments on the presentation of the paper.

## REFERENCES

- Ball M., Golden B., Assad A. and Bodin L. (1983). Planning for truck fleet size in the presence of a common carrier option. Decision Sciences 14, 103-120.
- Bodin L., Golden B., Assad A. and Ball M. (1983). The state of the art in the routing and scheduling of vehicles and crews. Computers and Operations Research 10, 63-212.
- Chien T. W. (1987). Solving Two Integrated Logistical Problems Using Lagrangian Relaxation Approaches. Ph.D. Dissertation, Krannert Graduate School of Management, Purdue University, West Lafayette, IN.
- Cornuejols G., Fisher M. and Nemhauser G. (1977). Location of bank accounts to optimize float: An analytic study of exact and approximate algorithms. Management Science 23, 789-810.
- Dantzig G. (1963). Linear Programming and Extensions. Princeton University Press, Princeton, NJ.
- Etschmaier M. and Mathaisel D. (1984). Aircraft scheduling: The state of the art. AGIFORS XXIV, 181-209.
- Etschmaier M. and Richardson R. (1973). Improving the Efficiency of Benders' Decomposition Algorithm for a Special Problem in Transportation. Technical Report No. 14, Department of Industrial Engineering, University of Pittsburgh, Pittsburgh, PA.
- Fisher M. (1981). The lagrangian relaxation method for solving integer programs. Management Science 27, 1-18.
- Fisher M., Jaikumar R. and Van Wassenhove L. (1980). A Multiplier Adjustment Method for the Generalized Assignment Problem. Decision Science Working Paper, University of Pennsylvania, Philadelphia, PA.
- Geoffrion A. (1974). Lagrangian relaxation and its use in integer programming. Mathematical Programming Research 2, 82-114.
- Golden B. and Assad A. (1986). Perspectives on vehicle routing: Exciting new developments. Operations Research 34, 803-810.
- Held M. and Karp R. (1970). The traveling salesman problem and minimum spanning trees. Operations Research 18, 1138-1162.
- Held M., Wolfe P. and Crowder H. P. (1974). Validation of subgradient

- optimization. Mathematical Programming 6, 62-88.
- Kennington J. and Helgason R. (1980). Algorithms for Network Programming. Wiley, New York.
- Kushige T. (1963). A solution of most profitable aircraft routing. AGIFORS III.
- de Lamotte H. and Mathaisel D. (1983). Experience with MPSX-MIP for Olympic Airways fleet assignment problems. MIT Flight Transportation Laboratory Memorandum.
- Lawler E. (1976). Combinatorial Optimization: Networks and Matroids. Holt, Rinehart and Winston, New York.
- Miller R. (1967). An optimization model for transportation planning. Transportation Research 1, 271-286.
- Richardson R. (1976). An optimization approach to routing aircraft. Transportation Science 10, 52-71.
- Shapiro J. (1979). A survey of Lagrangian techniques for discrete optimization. Annals of Discrete Mathematics 5, 113-138.
- Walker-Powell A. (1970). The port-linkage problem. Presented at the AGIFORS Study Group on Scheduling Construction and Evaluation, Vienna.

## Captions

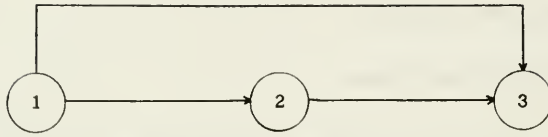
Table 1: Computational Results for Test Problems

Table 2: Breakdown of Total CPU Time

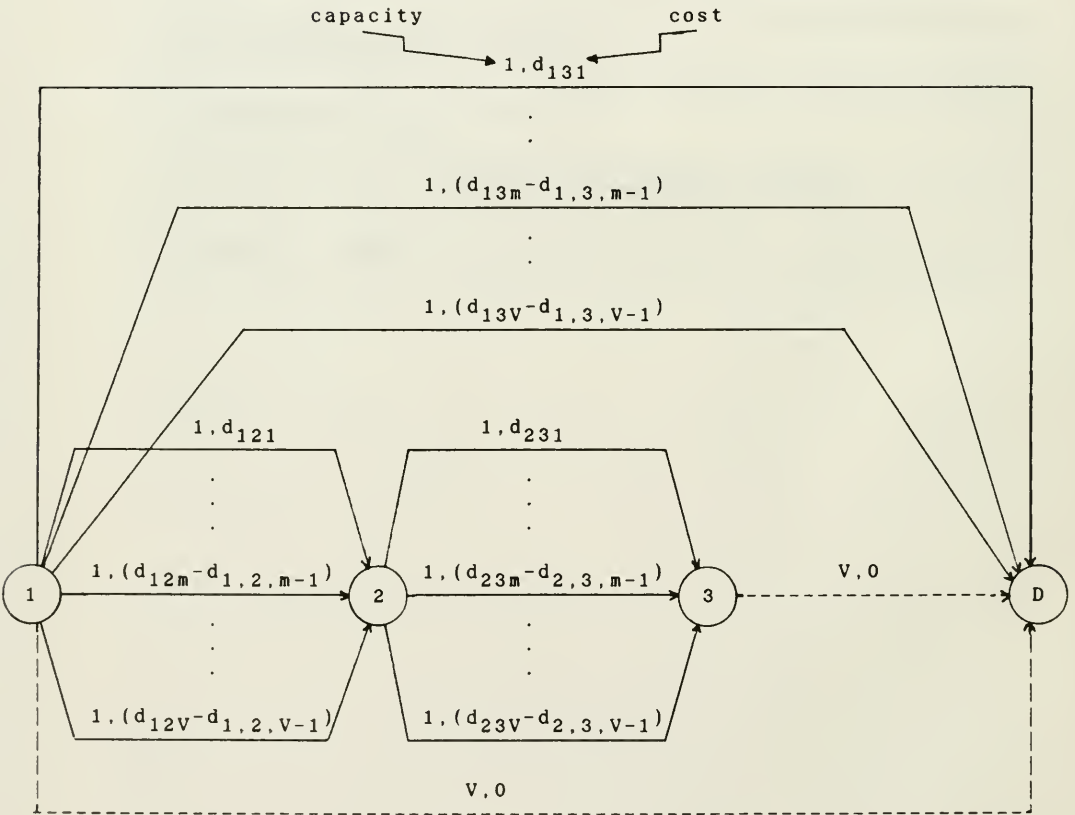
Table 3: Comparison of Solutions for the 12-City Test Problem

Fig. 1: Transforming [CARSS] into a Minimum-Cost Flow Problem

Figure 1



Original Network



Modified Minimum-Cost Network

Table 1

1 Aircraft

		<u>No. of Nodes</u>			
		<u>17</u>	<u>20</u>	<u>23</u>	<u>26</u>
Initial % GAP	max	58.76	55.86	62.38	64.14
	min	38.54	46.77	51.76	54.76
	avg	41.82	51.32	54.87	56.58
Final % GAP	max	2.17	1.89	1.21	1.28
	min	0.94	0.97	0.98	1.04
	avg	1.66	1.23	1.09	1.18
CPU secs* (Total)	max	74.87	142.53	148.23	200.42
	min	61.02	93.67	134.17	184.44
	avg	68.09	102.14	140.48	192.49

2 Aircraft

		<u>No. of Nodes</u>			
		<u>17</u>	<u>20</u>	<u>23</u>	<u>26</u>
Initial % GAP	max	56.13	54.38	46.23	58.54
	min	43.87	50.22	40.38	56.27
	avg	50.76	52.48	43.23	57.44
Final % GAP	max	3.22	2.42	2.58	2.62
	min	1.32	1.82	1.39	1.83
	avg	2.11	2.12	1.86	2.22
CPU secs* (Total)	max	141.31	226.71	289.01	363.14
	min	130.36	185.99	249.77	342.63
	avg	136.78	202.64	269.77	355.26

(continued on next page)

Table 1  
(continued)

3 Aircraft

		<u>No. of Nodes</u>			
		<u>17</u>	<u>20</u>	<u>23</u>	<u>24</u>
Initial	max	48.66	52.14	50.36	52.71
% GAP	min	45.27	43.03	47.29	41.15
	avg	46.35	48.11	48.91	45.14
Final	max	1.27	2.04	1.66	2.63
% GAP	min	1.09	1.03	1.02	1.03
	avg	1.18	1.48	1.38	1.76
CPU secs *	max	174.38	243.77	322.38	341.17
(Total)	min	142.37	223.86	287.66	312.38
	avg	158.42	234.46	304.37	328.79

4 Aircraft

		<u>No. of Nodes</u>		
		<u>17</u>	<u>20</u>	<u>23</u>
Initial	max	58.76	62.14	60.78
% GAP	min	48.65	56.71	53.23
	avg	52.17	58.29	56.44
Final	max	1.57	2.09	2.59
% GAP	min	1.02	1.54	1.83
	avg	1.39	1.82	2.20
CPU secs *	max	209.38	298.67	382.77
(Total)	min	184.23	267.14	338.13
	avg	196.62	277.94	355.72

Legend:

Initial % GAP = (initial UB - best LB)/(best LB)

Final % GAP = (final UB - best LB)/(best LB)

\* CPU times (excluding I/O) in seconds on IBM 3083.  
Each problem class contains 5 different problem instances.

Table 2

1 Aircraft

	<u>No. of Nodes</u>			
	<u>17</u>	<u>20</u>	<u>23</u>	<u>26</u>
UB <sup>1</sup> (excl. SP <sup>2</sup> )	67.2%	63.8%	61.2%	58.0%
SP	20.6%	21.4%	22.1%	23.7%
FEAS <sup>3</sup>	12.2%	14.8%	16.7%	18.3%

2 Aircraft

	<u>No. of Nodes</u>			
	<u>17</u>	<u>20</u>	<u>23</u>	<u>26</u>
UB(excl. NETFLO <sup>4</sup> )	53.2%	49.9%	47.6%	43.3%
NETFLO	30.4%	32.5%	33.1%	35.0%
FEAS	16.4%	17.6%	19.3%	21.7%

3 Aircraft

	<u>No. of Nodes</u>			
	<u>17</u>	<u>20</u>	<u>23</u>	<u>24</u>
UB(excl. NETFLO)	48.2%	46.5%	44.3%	40.2%
NETFLO	31.5%	31.8%	32.6%	34.5%
FEAS	20.3%	21.7%	23.1%	25.3%

4 Aircraft

	<u>No. of Nodes</u>		
	<u>17</u>	<u>20</u>	<u>23</u>
UB(excl. NETFLO)	44.2%	39.5%	38.5%
NETFLO	32.6%	34.8%	35.1%
FEAS	23.2%	25.7%	26.4%

1. UB: subgradient iteration
2. SP: shortest path subproblem
3. FEAS: feasible solution construction
4. NETFLO: minimum-cost flow subproblem

Table 3

Solution from Etschmaier and Richardson (1973)

Optimal Route: 1-3-5-8-9-12

Profit: 10312

Passenger Distribution:

	3	5	8	9	12
1	60	11	0	0	0
3		79	0	0	0
5			41	49	0
8				10	31
9					59

Solution obtained using Lagrangian Relaxation Algorithm:

Best Route: 1-2-3-5-8-9-12

Profit: 11390 (UB = 11451.94, %GAP = 0.54%, CPU\* = 16.83)

Passenger Distribution:

	2	3	5	8	9	12
1	26	60	4	0	0	0
2		0	7	0	0	0
3			79	0	0	0
5				41	49	0
8					10	31
9						59

\*CPU time in seconds on IBM 3083









Date Due

FEB 19 1992

MIT LIBRARIES DUPL 1



3 9080 00570432 2

