Journal of Combinatorial Optimization 5, 211–228 (1997)

A GRASP for Aircraft Routing in Response to Groundings and Delays

MICHAEL F. ARGÜELLO arguello@mail.utexas.edu
JONATHAN F. BARD* jbard@mail.utexas.edu
Graduate Program in Operations Research, Department of Mechanical Engineering, University of Texas, Austin, Texas 78712-1063, USA

GANG YU yu@uts.cc.utexas.edu
Department of Management Science and Information Systems and Center for Management of Operations and Logistics Graduate School of Business, University of Texas Austin, Texas 78712-1175, USA

Received September 5, 1996; Revised April 4, 1997

Abstract. This paper presents a greedy randomized adaptive search procedure (GRASP) to reconstruct aircraft routings in response to groundings and delays experienced over the course of the day. Whenever the schedule is disrupted, the immediate objective of the airlines is to minimize the cost of reassigning aircraft to flights taking into account available resources and other system constraints. Associated costs are measured by flight delays and cancellations.

In the procedure, the neighbors of an incumbent solution are generated and evaluated, and the most desirable are placed on a restricted candidate list. One is selected randomly and becomes the incumbent. The heuristic is polynomial with respect to the number of flights and aircraft. This is reflected in our computational experience with data provided by Continental Airlines. Empirical results demonstrate the ability of the GRASP to quickly explore a wide range of scenarios and, in most cases, to produce an optimal or near-optimal solution.

Keywords: GRASP, airline scheduling, real-time control, irregular operations

1. Introduction

Airlines spend a great deal of effort developing flight schedules for each of their fleets. A flight schedule consists of the originating city, departure time, destination, and arrival time for flights that the airline intends to serve. Due to the seasonality of passenger travel, flight schedules are created every two to three months. In the short term, however, aircraft are typically assigned to cover specific flights every day for a one week time horizon. The ordered sequence of flights to which an aircraft is assigned is called an aircraft route. A collection of aircraft routes that uses all available aircraft to service scheduled flights is defined as an aircraft routing. The great expense of purchasing and maintaining aircraft motivates airlines to maximize their utilization by creating aircraft routings with as little embedded idle time as possible. Thus when an aircraft is unexpectedly grounded or delayed, an airline's ability to fulfill its flight schedule is compromised.

*This work was partially supported by the Texas Higher Education Coordinating Board under grant ARP-003.

Unplanned aircraft shortages and resulting flight schedule disruptions are an unavoidable occurrence in the daily operations of an airline. Aircraft are grounded or temporarily delayed when equipment failures make flying unsafe, when severe weather closes an airport, or when the required flight crews are unavailable. Flights that are grounded or delayed jeopardize their assigned routes. Real-time decisions must be made to minimize lost revenues, passenger inconvenience, and operational costs by reassigning available aircraft and canceling or delaying flights.

In extreme cases where severe weather closes airports in a region, aircraft are prohibited from taking off or landing at the affected airports. This results in massive flight cancellations whose effects ripple through the system causing missed connections and other disruptions at upstream and downstream airports. Accordingly, an airline must be able to modify its aircraft routing in order to minimize the effects of extraordinary irregular events like severe weather. The airline must also be able to get back on schedule as quickly as possible after the irregular events have passed and the entire system is open. For example, if a hub like Newark with about 200 daily departing flights is closed due to a blizzard, and if each departing flight averages $30,000 in revenue, then up to $3 million at the hub airport alone could be saved if the airlines can get back on schedule in half a day instead of a full day. The results would be even more dramatic if we considered the revenue that could be saved by re-routing aircraft at open airports.

Due to the complexity of this problem, operations personnel are unable in practice, to evaluate alternate solutions with respect to these objectives and are content merely to find feasible solutions. In this paper we present a heuristic that quickly generates cost-effective aircraft routings in response to schedule disruptions.

Limited work has been published on this subject. Teodorovic and Guberinic (1984) present a branch and bound procedure to minimize total passenger delay. Teodorovic and Stojkovic (1990) give a lexicographic dynamic programming scheme to minimize the number of canceled flights and the total passenger delay. Neither of these papers considers flight delay and cancellation costs. The chief concern is for total passenger delay and number of flight cancellations.

Jarrah et al. (1993) introduce two minimum cost network flow models. One addresses flight delays and the other flight cancellations. Unfortunately, delays and cancellations cannot be considered simultaneously in these models. Yu (1995, 1996) also provides a network representation of the problem and derives an optimization model from it that appears to be NP-hard. No solution techniques are proposed. His ideas serve as a foundation for the work presented in Argüello (1997) where more formal optimization models are constructed and a greedy randomized adaptive search procedure (GRASP) is introduced.

A related problem arises from the ground delay program which is an FAA imposed reduction in capacity at airports suffering from inclement weather. For this problem, a reduced number of flight arrivals and departures are mandated for a specific period of time. The designation of flight delays and cancellations are determined locally with respect to the airport where the ground delay program is imposed. Yu and Luo (1997) consider a variety of objectives for the ground delay program as well as scenarios where resources may and may not be reassigned to other scheduled flights. They present efficient algorithms to solve some simple cases, prove the NP-hardness of some difficult cases, and provide heuristics.

Vasquez-Marquez (1991) considers the options available for shifting flight arrivals and gives a traveling salesman problem (TSP) formulation (although it appears that only an assignment problem is being solved in practice). The model was implemented as part of a decision support system for American Airlines and provided substantial savings in direct operating expenses.

This paper develops a GRASP and provides empirical results for data associated with Continental Airlines' 757 fleet. The GRASP described here is adapted for use as a randomized neighborhood search technique. Neighbor generation operations are performed on pairs of aircraft routes from an incumbent aircraft routing and result in the creation of new aircraft routings which can then be evaluated. A set of the best local routings is stored and one is selected at random to become the next incumbent. This process is repeated until either a local minimum is encountered or a CPU run time limit is reached. When applied to recent 757 fleet data obtained from Continental Airlines, this method generated near optimal aircraft routings.

The next section gives more details about the problem under consideration. Section 3 presents the GRASP along with the neighborhood definitions and the heuristic complexity. A lower bounding procedure is summarized in Section 4. In Section 5, the data and empirical results are presented. Conclusions are offered in Section 6.

2. Problem definition

To date, there are no comprehensive automated systems in use at commercial airlines that resolve the irregular operations aircraft routing problem. Typically, when aircraft are grounded or delayed, operations managers query databases containing flight schedule, aircraft routing, aircraft maintenance, and crew schedule information. Armed with this data, operations managers consider aircraft fleets separately in their attempt to find simple routing alternatives that put the airline back on schedule. The guiding rule is to find the simplest alternative with the least schedule disruption. The most attractive candidates are those that contain the least number of canceled flights. The routing alternative that minimizes disruptions is the one that incurs the least cost in terms of passenger inconvenience or lost flight revenues. A routing alternative that gets the airline back on schedule is one that positions aircraft at the end of the recovery period so that the original flight schedule can then be resumed.

When operations managers search for attractive routing alternatives, they do not consider modifying the route of an aircraft that requires maintenance service during the immediate recovery period. They also do not consider explicitly the crew requirements to staff candidate routings. When a routing is selected, it is then passed to the crew coordinators for a crew feasibility review. Thus new routings do not alter the aircraft maintenance schedule during the recovery period and are acceptable only if appropriate personnel can be found to fly the aircraft.

2.1. Constraints

The problem that this paper addresses closely reflects the problem that must be solved in practice. The objective is to minimize the flight cancellation and delay costs associated with a recovery aircraft routing in response to groundings and delays. Every aircraft routing considered must be feasible with respect to the following constraints: (1) every flight in each aircraft route must depart from the station where the immediately preceding flight arrived; (2) a minimum turnaround time must be enforced between each flight arrival and subsequent departure; (3) the recovery period extends to the end of the current day; (4) aircraft must be positioned at the end of the recovery period so that the flight schedule can be resumed the next day; (5) station departure curfew restrictions will be observed; and (6) no aircraft scheduled for maintenance service during the recovery period will have its original route altered.

To evaluate aircraft routings, cancellation and delay costs must be given with the flight schedule. The cost of any alternate schedule is composed of the cost of each aircraft route and the sum of the cancellation costs for every canceled flight. The cost of an aircraft route is determined by observing the delay that flights will incur in the route and summing the costs for each of those delays. The first constraint above implies that the flow must be continuous and that no ferrying—transporting empty aircraft to another location—is permitted. In practice, ferrying aircraft is an option of last resort and is to be avoided as much as possible. The third constraint is somewhat arbitrary because we would like to recover as soon as possible. If we cannot get back on schedule by the end of the day, it is an easy matter to extend the recovery period to the following day or beyond. The fourth constraint is enforced by requiring specific quantities of aircraft to be located at individual airports by the end of the recovery period. This constraint is defined as aircraft balance. In keeping with airline practice, aircraft with maintenance scheduled during the recovery period will not be considered for route modifications. In addition, only single fleets will be considered, and crew availability will not affect the generation of alternate routings.

2.2. Mathematical model

In Argüello (1997) a series of alternative models is introduced to represent the irregular operations problem. The resource assignment model presented below captures the details of the problem in the most concise and intuitive manner. In the formulation, aircraft are resources that are assigned to aircraft routes that are implicitly feasible with respect to the first, second, third, and fifth constraints noted above. The fourth constraint is enforced explicitly.

The following notation is used in the formulation.

Indices

- i flight index
- j aircraft route index
- k aircraft index
- t station index

Sets

- F set of flights
- P set of feasible aircraft routes
- Q set of available aircraft
- S set of stations

Parameters

a_{ij} equal to 1 if flight i is in aircraft route j; 0 otherwise
b_{tj} equal to 1 if aircraft route j terminates at station t; 0 otherwise
c_i cost of canceling flight i
d_j^k cost for assignment of aircraft k to aircraft route j
h_t number of aircraft required to terminate at station t

Variables

x_j^k assignment of aircraft k to aircraft route j
y_i assignment of cancellation to flight i

Resource assignment mathematical formulation

$$\text{minimize} \quad \sum_{k\in Q}\sum_{j\in P} d_j^k x_j^k + \sum_{i\in F} c_i y_i \tag{1a}$$

subject to

$$\text{(flight cover)} \quad \sum_{k\in Q}\sum_{j\in P} a_{ij} x_j^k + y_i = 1 \quad \forall i \in F \tag{1b}$$

$$\text{(aircraft balance)} \quad \sum_{k\in Q}\sum_{j\in P} b_{tj} x_j^k = h_t \quad \forall t \in S \tag{1c}$$

$$\text{(resource utilization)} \quad \sum_{j\in P} x_j^k = 1 \quad \forall k \in Q \tag{1d}$$

$$\text{(binary aircraft assignment)} \quad x_j^k \in \{0, 1\} \quad \forall j \in P, k \in Q \tag{1e}$$

$$\text{(binary cancellation assignment)} \quad y_i \in \{0, 1\} \quad \forall i \in F \tag{1f}$$

The sets F, Q and S, and parameters c_i and h_t are inputs derived from the flight schedule, aircraft routing, and flight cancellation costs. Set P is the union of all feasible aircraft routes including null routes commencing from every airport where aircraft are available. Parameters a_{ij} and b_{tj} are determined by inspecting each aircraft route in set P, and d_j^k is calculated by observing the delay that flights will incur when aircraft resource k is assigned to aircraft route j and summing the costs for each of those delays. It should be noted that $d_j^k = \infty$ for all aircraft routes j that commence at a station other than where aircraft resource k is located.

In the model, the objective function (1a) minimizes the sum of the assigned aircraft route costs and the flight cancellation costs. The flight cover constraint (1b) stipulates that all flights must be either in an assigned aircraft route or canceled. The aircraft balance constraint (1c) implements the requirement of positioning aircraft at the end of the recovery period so that the flight schedule can then commence as planned. The resource utilization constraint (1d) ensures that each aircraft is assigned to exactly one feasible aircraft route. The binary assignment constraints (1e), (1f) reflect the reality that fractional aircraft cannot be assigned to aircraft routes and flights cannot be partially canceled.

The irregular operations problem under consideration can be modeled accurately with the resource assignment model (1); however, it is extremely difficult to solve. The number of feasible aircraft routes is exponential with respect to the number of flights thus making enumeration intractable. In addition, this model is a general integer program for which there does not appear to be any special structure that can be exploited. It may be possible that column generation techniques may be useful in generating feasible aircraft routings which can then be evaluated, but it is not clear that such methods will be useful in practice.

In an attempt to generate feasible aircraft routings, a GRASP has been developed for this problem. The details are presented in the next section.

3. Randomized search heuristic

GRASPs have been used to find high quality solutions to a variety of logistics and combinatorial optimization problems including maintenance base planning (Feo and Bard, 1989), machine scheduling (Feo et al., 1991), and number partitioning (Argüello et al., 1996) to name a few. Like most heuristics, GRASP has two basic components, a solution construction phase and a local search phase. The former uses a greedy evaluation function and a randomized selection method to iteratively construct a feasible solution. At each iteration, alternate moves are generated and evaluated. The most favorable are stored on a restricted candidate list (*rcl*) from which one is then selected at random. Membership on the *rcl* is limited to preserve the integrity of the greedy evaluation function. The *rcl* may be restricted by cardinality or quality. For example, the best X moves or all moves within Y percent of the best move may be stored in the *rcl*. This process is repeated until a feasible solution is obtained. This constitutes one iteration.

The local search phase uses the solution from the construction phase to find a local minimum usually with respect to an exchange argument. In general, the first phase employs randomization to select from a set of moves that are favorable with respect to a greedy evaluation function while the second phase uses a deterministic exchange procedure to attain a local minimum. The balance between the greedy evaluation function and the randomized selection procedure as controlled by the *rcl* in the construction phase implies that many repetitions of the GRASP can result in a variety of high quality solutions.

When considering aircraft groundings and delays, it is important to note that the solution obtained by simply canceling flights to which affected aircraft were assigned may be feasible or near feasible. In fact, this solution can be infeasible with respect to only one constraint: aircraft balance. When this is the case and the solution can be modified so that it is feasible with respect to aircraft balance, even at the expense of the airport curfew constraint, then the new solution can be used as a starting point in a neighborhood search procedure. If the starting solution happens to violate the airport curfew constraint, a large penalty cost can be assigned to it in order to encourage the greedy neighborhood search procedure to move to a more attractive neighbor. The advantages of using this initial solution are that it is exactly or close to the original aircraft routing that the airline planned to use and that the effort required to construct a feasible aircraft routing may not be trivial.

The heuristic presented here is a neighborhood search technique that incorporates the basic components of GRASP. The initial feasible solution is used as input for the heuristic, so the construction phase is unnecessary and omitted. Instead the greedy evaluation function and randomized selection method are used in the local search phase to find promising feasible solutions. As these are generated, they are evaluated with respect to cost and the

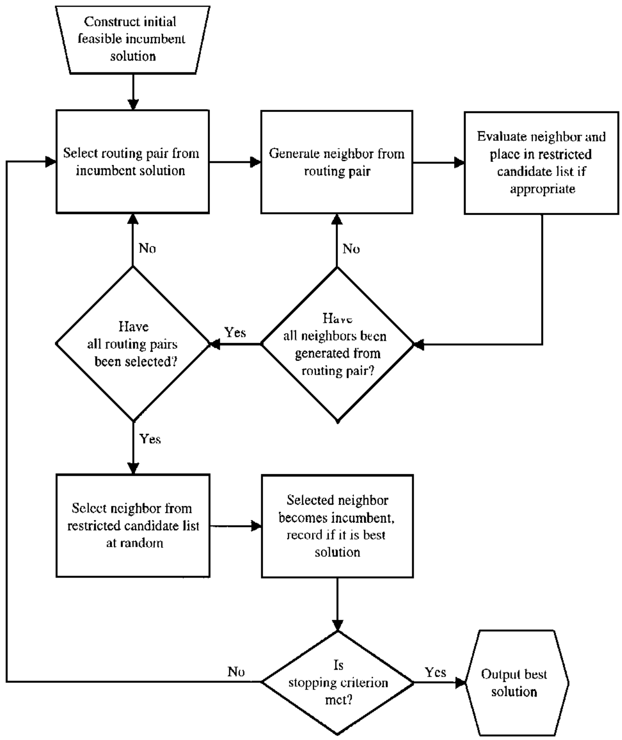

Figure 1. GRASP flow diagram.

most desirable are placed on the *rcl*. One is selected randomly from the list and becomes the incumbent for the next iteration. Thus the local search phase acts as a randomized search procedure. The computations are repeated until one of several stopping criteria are met. The major components of the heuristic are the use of a feasible solution as an incumbent solution, the generation and evaluation of the incumbent's neighboring solutions, the storing of the most desirable solutions in a restricted candidate list, the randomized selection of a neighboring solution as incumbent for the next iteration, and the repetition of the procedure until a stopping criterion is met. A flow diagram of the GRASP is presented in figure 1. The individual steps are discussed presently.

As conceived, the GRASP is analogous to a controlled random walk beginning at the original incumbent solution and moving to each of the selected neighbors. It is controlled in the sense that only the most desirable neighbors are candidates to become incumbents. The entire process may be repeated so that many paths are explored.

In many ways, our heuristic resembles an implementation of tabu search (Glover, 1990) or simulated annealing (Kirkpatrick et al., 1983). Nevertheless, there are basic components associated with these paradigms that are not included here. For example, no memory functions or tabu move designations inherent to tabu search are used by the heuristic. A temperature function critical to overcoming local minima in simulated annealing is not found in the heuristic either.

The idea of generating solutions from an incumbent solution suggests that a genetic algorithm might be a promising method for solving the problem under consideration. Indeed, the authors have developed a genetic algorithm that uses an efficient representation of aircraft routings, defines a crossover operation to be performed on pairs of routing representations, and implements a reparation procedure to construct feasible aircraft routes in the child routings. This is undoubtedly a novel method for generating solutions for the problem at hand. However, the GRASP presented here is more attractive because of the balance of the greedy evaluation function and the randomized selection procedure and the limited CPU time needed to generate high quality solutions. It is our conjecture that a genetic algorithm with a completely random crossover operation will require significantly more CPU time to generate the solutions that the GRASP is capable of producing quickly. Nevertheless, it may be interesting to investigate the genetic algorithm at some later date.

3.1. Neighborhood definitions

An aircraft routing consists of a set of routes that uses all available aircraft to service scheduled flights. As defined previously, an aircraft route is an ordered sequence of flights to which an aircraft is assigned. To deal with flight cancellations we introduce the idea of a cancellation route. Cancellation routes can be constructed from the set of canceled flights so that each cancellation route is an ordered sequence of flights that could be assigned to an actual aircraft without requiring ferrying between the flights. It may be possible that a cancellation route has only one flight. An aircraft routing may now be expanded to consist of a set of aircraft routes to which available aircraft may be assigned and a set of cancellation routes that contains all canceled flights. Thus an aircraft routing consists of a set of routes, some of which are aircraft routes and the rest of which are cancellation routes.

A *route pair* is defined to be either a pair of aircraft routes or one aircraft route paired with one cancellation route. Route pairs are used in the GRASP to construct neighboring solutions in which all the routes other than those in the route pair are identical and a newly generated route pair replaces the one under consideration.

Three operations are used to generate neighboring solutions: *flight route augmentation*, *partial route exchange*, and *simple circuit cancellation*. The first two are performed on route pairs and the third is performed on individual aircraft routes. The purpose of these operations is to construct feasible solutions that can serve as alternatives to the incumbent.

Table 1 lists two sample routes, each consisting of five flights. Route 1 begins with flight 101 from ORF (Norfolk, VA) to EWR (Newark, NJ) and ends with flight 105 from BDL (Hartford, CT) to CLE (Cleveland, OH). Route 2 originates in CLE with flight 201 and terminates in MDW (Chicago, Midway Airport) after the completion of flight 205. Figure 2(a) illustrates these routes. They will be used to illustrate the neighborhood construction operations.

The flight route augmentation operation on a route pair removes a flight or sequence of flights from one route and places these flights in the other route. The second or target route is augmented by the flights removed from the first or source route. The target route can be augmented in three distinct ways. First, a circuit can be placed in front of it. A circuit is a sequence of flights that originates and terminates at the same airport. An example of this

Table 1. Sample route pair.

Route	Flight	Origin	Destination
	101	ORF	EWR
	102	EWR	STL
1	103	STL	CLE
	104	CLE	BDL
	105	BDL	CLE
	201	CLE	ATL
	202	ATL	EWR
2	203	EWR	BWI
	204	BWI	CLE
	205	CLE	MDW

kind of augmentation involves removing flights 104 and 105 from route 1 and relocating them in front of route 2. The resulting route pair (see figure 2(b)) is route 1 (101-102-103) and route 2 (104-105-201-202-203-204-205).

A second way to augment a target route is to insert a circuit between its first and last flights. A circuit may be placed in a route wherever its starting and ending airport intersects a point in the route. For example, the circuit consisting of flights 104 and 105 may be removed from route 1 and placed after flight 204 in route 2. This results in route 1 (101-102-103) and route 2 (201-202-203-204-104-105-205) as shown in figure 2(c). Note that both of these augmentation methods preserve the origination and termination airports for each route in the route pair. The third augmentation method allows for the exchange of termination airports when the route pair is a pair of aircraft routes. This augmentation consists of removing a sequence of flights from the source route and then appending it to the end of the target route. This augmentation may be performed only on a pair of aircraft routes so that aircraft balance (constraint 4) can be preserved. An example of this operation is the removal of flight 205 from route 2 and placing it at the end of route 1 so that route 1 (101-102-103-104-105-205) and route 2 (201-202-203-204) exchange their destinations (see figure 2(d)). Aircraft balance is retained because the incumbent is assumed to be feasible. However, if the source route is a cancellation route, any appending augmentation to an aircraft route must consist of a circuit to preserve the target route's termination requirement. Recall that only feasible neighbors are evaluated.

The partial route exchange operation is a simple exchange of a pair of flight sequences. Two kinds of exchanges are possible. The first is the exchange of flight sequences with identical endpoints. For example, flights 102 and 103 may be exchanged for flights 203 and 204. As shown in figure 2(e), this gives route 1 (101-203-204-104-105) and route 2 (201-202-102-103-105). The second type of exchange also results in an exchange in termination airports. The sequence of flights being exchanged must have the same origination airport, but the termination airports are then swapped. Exchanging flight 205 for flights 104 and 105 is an example of this kind of operation and leads to route 1 (101-102-103-205) and route 2 (201-202-203-204-104-105) as depicted in figure 2(f). This second type of exchange is

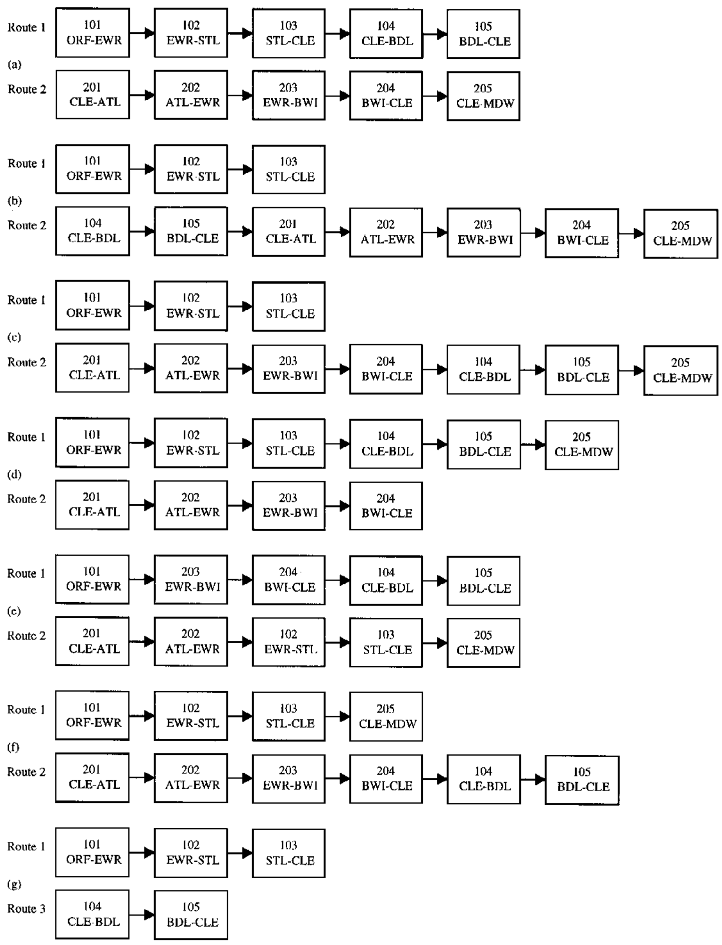

Figure 2. Sample route pair and neighbors.

limited to aircraft route pairs. A cancellation route may not exchange its termination airport with an aircraft route because this would result in an aircraft balance violation.

The flight route augmentation and partial route exchange operations are performed on a route pair. The simple circuit cancellation operation is performed only on one aircraft route. This operation simply removes a circuit from the aircraft route and places it in a newly created cancellation route. The motivation for this operation is to allow for the cancellation of a circuit of flights without having to find a cancellation route in which to fit it. The result is the cancellation of the circuit and the reduction of flights in the route from which the circuit is taken. To illustrate, create empty cancellation route 3 and place flights 104 and 105 in it. The result of this operation gives (figure 2(g)) route 1 (101-102-103) and route 3 (104-105). Only circuits are considered so that the remaining aircraft route is feasible with respect to flow continuity and aircraft balance.

Table 2. Feasible neighbor routes for sample route pair.

No.	Route 1	Route 2	Operation
1	101-203-204-104-105	201-202-102-103-205	Exchange
2	101-102-103-201-202-203-204-104-105	205	Augmentation
3	101-102-103-104-105-201-202-203-204	205	Augmentation
4	101-102-103-201-202-203-204	104-105-205	Exchange
5	101-203-204	201-202-102-103-104-105-205	Exchange
6	101-102-103-104-105-205	201-202-203-204	Augmentation
7	101-102-103	104-105-201-202-203-204-205	Augmentation
8	101-102-103	201-202-203-204-104-105-205	Augmentation
9	101-102-103-205	201-202-203-204-104-105	Exchange
10	101-203-204-205	201-202-102-103-104-105	Exchange
11	101-102-103-201-202-203-204-205	104-105	Exchange
12	101-102-103-104-105-201-202-203-204-205	Null	Augmentation

Table 2 presents all feasible neighbors for the sample routes. Both route 1 and route 2 are assumed to be aircraft routes. If either route is a cancellation route, then solutions 6, 9, 10, 11, and 12 would be infeasible due to their exchange of termination airports. Route 2 (null) in solution 12 refers to a route with no flights. It should be noted that the neighboring solutions are feasible with respect to both flow continuity (constraint 1) and aircraft balance (constraint 4). However, nothing can be said about airport curfew feasibility because flight departure and duration times and airport curfew times have not been stated.

3.2. *Heuristic complexity*

Beginning with the initial incumbent aircraft routing, the GRASP inspects all pairs of aircraft routes, all aircraft route-cancellation route pairs, and finally all individual aircraft routes in order to perform the neighbor generation operations. It evaluates all feasible neighbors and stores the best on a restricted candidate list. After all feasible neighbors are generated and evaluated, one is selected at random from the *rcl* to become the incumbent. This procedure is repeated until either an empty *rcl* is encountered or a CPU time limit is reached. The entire process may also be repeated from the initial incumbent an arbitrary number of times. Figure 3 presents the pseudocode for the GRASP.

The generation and evaluation of an aircraft routing's neighborhood can be done efficiently. Our implementation of the GRASP requires that for any incumbent solution, every neighboring solution must be generated and evaluated. To do this, each aircraft route must be compared with every other aircraft route and with every cancellation route. A solution with m aircraft routes and n cancellation routes will have $\frac{m(m-1)}{2}$ aircraft route pairs and $m \times n$ aircraft-cancellation route pairs. This implies that the effort required to enumerate every route pair is quadratic with respect to the $m + n$ routes in a solution. In addition, for any two routes of length p and q, respectively, the number of inspections required to generate all neighboring solutions is $O(p^2q^2)$, which is polynomial with respect to the

```
procedure GRASP iteration

input: incumbent_routing, best_routing

output: new_incumbent, best_routing

begin
let r = number of routes in incumbent_routing
for i = 1 to r - 1 do                                          /* select first route */
    begin
    select route i
    if route i is aircraft route let route1_type = A           /* define first route */
        else let route1_type = C
    let m = number of flights in route i
    for j = i + 1 to r do                                      /* select second route */
        begin
        select route j
        if route j is aircraft route let route2_type = A       /* define second route */
        else let route2_type = C
        if not (route1_type = C and route2_type = C)           /* cancellation pairs not considered */
            begin
            let n = number of flights in route j
            for u = 1 to m do                                  /* select starting point in first route */
                begin
                for v = u to m do                              /* select ending point in first route */
                    begin
                    for x = 1 to n do                          /* select starting point in second route */
                        begin
                        for y = x to n do                      /* select ending point in second route */
                            begin
                            if flights u through v is circuit and route2_type = A
                                begin
                                do flight route augmentation in front of first flight in route 2
                                evaluate feasible neighbor and place in rcl if appropriate
                                do flight route augmentation in front of flight x in route 2
                                evaluate feasible neighbor and place in rcl if appropriate
                                do flight route augmentation after last flight in route 2
                                evaluate feasible neighbor and place in rcl if appropriate
                                end
                            if flights x through y is circuit and route1_type = A
                                begin
                                do flight route augmentation in front of first flight in route 1
                                evaluate feasible neighbor and place in rcl if appropriate
                                do flight route augmentation in front of flight u in route 1
                                evaluate feasible neighbor and place in rcl if appropriate
                                do flight route augmentation after last flight in route 1
                                evaluate feasible neighbor and place in rcl if appropriate
                                end
                            if flights u and x have same origin and v and y same destination
                                begin
                                do partial route exchange with u-v and x-y
                                evaluate feasible neighbor and place in rcl if appropriate
                                end
                            if flights u and x have same origin and both routes are type A
                                begin
                                do partial route exchange with u-m and x-n
                                evaluate feasible neighbor and place in rcl if appropriate
                                end
                            if flights u through v is circuit and route1_type = A
                                begin
                                do simple circuit cancellation
                                evaluate feasible neighbor and place in rcl if appropriate
                                end
                            if flights x through y is circuit and route2_type = A
                                begin
                                do simple circuit cancellation
                                evaluate feasible neighbor and place in rcl if appropriate
                                end
                            end
                        end
                    end
                end
            end
        end
    end
new_incumbent = select neighbor at random from rcl
if cost(new_incumbent) < cost(best_routing), best_routing = new_incumbent
end
```

Figure 3. GRASP pseudocode.

number of flights in the routes. A route of length p only requires $O(p)$ time to evaluate. Thus the neighborhood of a solution can be generated and evaluated in polynomial time. This implies that each iteration can be implemented efficiently; however, the number of iterations necessary to find a global or even local minimum may be exponential. To avoid indefinite running of the GRASP, a time limit is included among the stopping criteria.

4. Lower bound procedure

To gauge the quality of the GRASP solutions, a lower bounding procedure has been developed for this problem. This procedure, referred to as the time-band approximation scheme, is detailed in Argüello (1997). It consists of creating a time-based minimum cost network flow problem in which the arc costs, corresponding to specific flight departures, estimate the delay costs associated with delayed flights. In fact, the arc costs underestimate the actual delay costs so that the optimal objective function value of the network flow model will always be no worse than the optimal solution for the problem considered here, thus providing a lower bound.

The first step in the approach is to create a network in which nodes represent time-based aircraft locations and arcs represent flights that have been scheduled for the time horizon under consideration. The nodes aggregate possible aircraft availability for a segment of time. The arc costs represent associated flight delay costs. This network discretizes the time horizon into uniform length time bands. The resulting model has an objective function that minimizes the cost of the flow throughout the network subject to conservation of flow, fight cover, and integral flow. The first set of constraints implicitly represents the aircraft utilization and aircraft balance constraints for the original problem. Thus the resulting formulation approximates the original problem as a standard minimum cost network flow problem with a flight cover side constraint. This is a general integer program for which special purpose solution algorithms do not currently exist. Relaxation of the integrality restriction yields a linear program that can be easily solved. The LP solution serves as the lower bound and thus can be used to gauge the quality of any feasible solution.

The time-band approximation scheme network transformation creates $O(s \cdot t)$ nodes and $O(f \cdot s \cdot t)$ arcs, where s is the number of airports, t is the number of time bands, and f is the number of flights. Although t is arbitrary, the network transformation is polynomial with respect to the number of airports and flights. The fact that the LP relaxation is derived directly from the time-based network implies that the lower bound can be determined efficiently.

5. Computational experience

The GRASP and lower bounding procedure were applied to a recent 757 flight schedule obtained from Continental Airlines. The schedule consists of 42 flights serviced by 16 aircraft over a network of 13 airports spanning eight time zones and three continents. The 757 is a long haul aircraft so the average flight time for each flight is five hours. The original aircraft routing consisted of 10 aircraft each assigned to two flights, two aircraft assigned to three flights, and four aircraft assigned to four flights. Table 3 presents the data from the flight schedule and aircraft routing. This data set is small enough to be able to inspect manually, yet large enough to test the performance of our procedures.

The cancellation costs in Table 3 represent lost profit as determined by a survey of the actual fares for the flights, assuming 75% passenger capacity and 10% profit margin. A

Table 3. Flight schedule for 757 fleet.

Flight	Origin	Destination	Departure†	Arrival†	Aircraft	Cancellation cost
170	LAX	EWR	660	970	101	6015
203	EWR	SAN	1040	1398	101	6435
239	EWR	LAX	660	1025	102	6015
184	LAX	EWR	1080	1385	102	6015
285	EWR	FLL	435	607	103	4020
392	FLL	EWR	655	823	103	4020
703	EWR	LIM	895	1356	103	5490
704	LIM	EWR	1435	1885	103	5490
711	EWR	IAH	630	858	104	5085
712	IAH	BOG	925	1212	104	6060
176	SAN	EWR	660	955	105	6435
197	EWR	SNA	1035	1395	105	5745
192	SEA	EWR	645	940	106	6585
189	EWR	SFO	1005	1385	106	6015
173	EWR	SFO	630	1005	107	6015
174	SFO	EWR	1080	1390	107	6015
150	EWR	IAH	360	583	108	5085
151	IAH	SFO	640	887	108	4965
488	SFO	IAH	940	1150	108	4965
225	IAH	LAX	1260	1467	108	4515
240	SFO	EWR	660	970	109	6015
195	EWR	SEA	1050	1405	109	6585
123	EWR	LAX	480	845	110	6015
133	LAX	IAH	934	1120	110	4515
134	IAH	EWR	1190	1386	110	5085
196	SNA	EWR	660	945	111	5745
1685	EWR	LAS	1130	1462	111	3555
1684	LAS	EWR	1545	1828	111	3555
233	EWR	MIA	455	636	112	4110
236	MIA	EWR	720	900	112	4110
63	EWR	LAX	965	1325	112	6015
186	LAX	EWR	1530	1830	112	6015
1641	EWR	LAS	465	797	113	3555

(*Continued*)

Table 3. (*Continued.*)

Flight	Origin	Destination	Departure†	Arrival†	Aircraft	Cancellation cost
1640	LAS	EWR	880	1160	113	3555
1643	EWR	LAS	1230	1556	113	3555
1642	LAS	EWR	1615	1898	113	3555
73	LAX	EWR	780	1095	114	6015
74	EWR	MAN	1200	1610	114	6540
75	MAN	EWR	370	830	115	6540
241	EWR	LAX	1065	1425	115	6015
709	BOG	IAH	540	845	116	6060
710	IAH	EWR	920	1120	116	5085

†Departure and arrival times are in minutes since midnight, eastern daylight time.

uniform delay cost of \$20 per minute of delay is imposed on all delayed flights. This is a conservative cost proposed originally in Jarrah et al. (1993).

A total of 6068 problem instances were generated by systematically grounding one through five aircraft at the beginning of the operations day. The groundings were selected at the beginning of the day along with the one day time horizon; the recovery period extends to the end of the day. All aircraft that are grounded are assumed to be available at the start of the next day at their current location. The aircraft balance constraint requires that aircraft must be re-routed so that the same number of aircraft are at each airport at the end of the day as when the day starts.

It should be noted that the instances tested did not include delayed aircraft. It was determined that the more difficult problem is the one in which resources are eliminated for the entire time horizon, so instances with several grounded aircraft were selected instead of a few grounded and a few delayed. No maintenance requirements were placed on any aircraft because their inclusion would not make the problem any more difficult.

After brief experimentation, the GRASP was implemented with an *rcl* cardinality limit of 10 neighbors with the proviso that none could exceed the incumbent by more than \$1000 in value. The procedure was allowed to run for a maximum of 2 CPU seconds and was restarted a total of five times per instance. Thus 10 CPU seconds were allocated for each instance.

The time-band approximation scheme was used with 30 minute time band lengths. The corresponding linear programs averaged 185 rows, 857 columns, and 2118 non-zero entries. Each was solved using the primal simplex method in CPLEX Version 4.3. On average, CPLEX required about one second to solve each instance.

Both the GRASP and the linear program interface to CPLEX were implemented with standard ANSI C programs. All programs were run on a SUN Sparcstation 10.

Table 4 presents the average cost for the time-band lower bound, the best GRASP solution, the solution obtained by simply canceling grounded aircraft flights, and the initial incumbent solution for the 6068 instances tested. We observed that the third solution type was infeasible with respect to aircraft balance 5852 times. This necessitated the generation of an initial incumbent solution that would be feasible with respect to this constraint. The fourth solution

Table 4. Solution type comparison.

Solution type	Average cost	Feasible solutions
Time-band	51641	0
GRASP	53647	6068
Cancel	62781	216
Initial	69748	5656

Table 5. Solution comparison to lower bound for 6068 instances.

Solution type	Optimum	5%	10%	15%	20%
GRASP	352 (5.8%)	4355 (71.8%)	5489 (90.5%)	5864 (96.6%)	6018 (99.2%)
Cancel	49 (0.8%)	160 (2.6%)	482 (7.9%)	1386 (22.8%)	2634 (43.4%)
Initial	18 (0.3%)	268 (4.4%)	438 (7.2%)	720 (11.9%)	1182 (19.5%)

type, however, was infeasible with respect to airport curfew 412 times. This infeasibility did not adversely affect the performance of the GRASP, which used the initial incumbent as the starting point. Figure 4 illustrates the relative value of the four solution types. The GRASP, while not attaining the lower bound, generated solutions far superior to doing nothing; i.e., simply canceling grounded aircraft flights, and to simply finding aircraft balance feasibility. Also, the GRASP always produced feasible solutions.

Table 5 presents the results of the GRASP and other two solution types compared to the lower bound. Of the 6068 instances tested, the best GRASP solution matched the time-band lower bound value 352 times. Thus for 5.8% of the instances a provably optimal

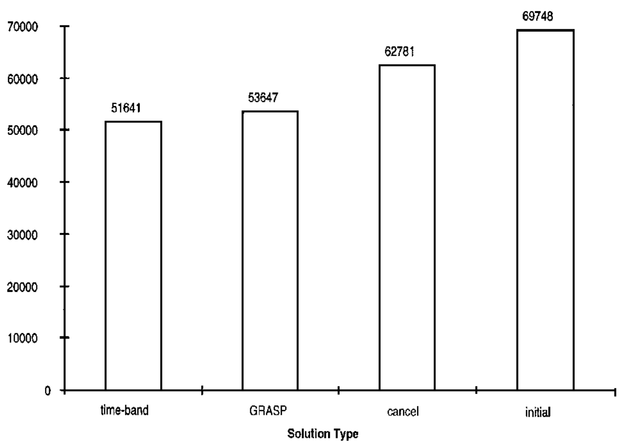

Figure 4. Comparison of average solution costs.

solution was obtained. The best GRASP solution was within 5% of the lower bound in 4355 instances. Over 90% of the instances had the best GRASP solution within 10% of the lower bound. Finally, 96.6% and 99.2% of the time the best GRASP solution was within 15% and 20% of the lower bound value, respectively. The GRASP solution offers a great improvement over the other two solution types.

6. Summary and conclusions

This paper has presented a randomized neighborhood search procedure to generate aircraft routings in response to groundings and delays. In the methodology neighboring solutions are derived from an incumbent by comparing aircraft and cancellation route pairs. The neighbor generation operations, *flight route augmentation*, *partial route exchange*, and *simple circuit cancellation*, are applied to route pairs and single aircraft routes to enumerate feasible neighbor solutions. The best of the neighbor solutions are stored on a restricted candidate list from which one is selected at random and becomes the new incumbent. This procedure is repeated until one of several stopping criteria is encountered, and is implemented in polynomial time.

As a complement to the GRASP, the time-band approximation scheme is used to obtain a lower bound to the problem under consideration. This scheme transforms the aircraft routing problem into a time-based network in which the time horizon is discretized into time bands. The corresponding mathematical model is an integral minimum cost network flow problem with a set of flight cover side constraints. The linear program resulting from relaxing the integrality constraints is solved to produce a lower bound.

The application of the GRASP and the lower bounding procedure to 757 fleet data obtained from Continental Airlines produces encouraging results. The best GRASP solution is provably very good and sometimes optimal. The quality of the lower bounding procedure has not been determined. It is possible that the lower bounds obtained are loose enough so that a significant number of GRASP solutions that do not match the lower bound value may indeed be optimal as well. Nevertheless, for over 90% of the instances tested, the best GRASP solution was within 10% of the lower bound, and the GRASP and lower bounding procedure required less than 15 CPU seconds per instance. This is extremely important for real-time decision making.

Although the GRASP proved effective with the 757 fleet data, its application to larger data sets would seem to be logical for future work. It would be interesting to see if solution quality deteriorates as a function of the number of scheduled flights or grounded aircraft. It would also be interesting to see how the *rcl* parameters, cardinality and threshold limits, affect solution quality for larger data sets. Also of interest would be the improvement in solution quality if at each iteration the best solution found was saved along with the new incumbent. Currently the GRASP returns the best incumbent solution, but it also could produce the best overall solution visited, even if it never became an incumbent. Finally, of special interest would be the relationship between running time and improvements in solution quality.

Further investigations of the time-band approximation scheme also seem to be warranted. A characterization of the tightness of the lower bound would provide better insight on the quality of feasible solutions. Attempts should also be made to solve the integral minimum cost network flow problem resulting from the network transformation of the problem. Certainly, the objective function value of an integer solution could not be worse than that of the lower bound, thus a tighter lower bound may be obtained. An integer solution could then be transformed into an aircraft routing whose actual value can be evaluated. It is possible that the corresponding aircraft routings will themselves be of high quality with respect to the lower bound. We have solved a few instances and found this to be the case.

References

M.F. Argüello, "Framework for exact solutions and heuristics for approximate solutions to airlines' irregular operations control aircraft routing problem," Ph.D. Dissertation, Department of Mechanical Engineering, University of Texas, Austin, 1997.

M.F. Argüello, T.A. Feo, and O. Goldschmidt, "Randomized mthods for the number partitioning problem," *Computers & Operations Research*, vol. 23, no. 2, pp. 103–111, 1996.

T.A. Feo and J.F. Bard, "Flight scheduling and maintenance base planning," *Management Science*, vol. 35, no. 12, pp. 1415–1432, 1989.

T.A. Feo, J.F. Bard, and K. Venkatraman, "A GRASP for a difficult single machine scheduling problem," *Computers & Operations Research*, vol. 18, no. 8, pp. 635–643, 1991.

F. Glover, "Tabu search: A tutorial," *Interfaces*, vol. 20, no. 4, pp. 74–94, 1990.

A.I.Z. Jarrah, G. Yu, N. Krishnamurthy, and A. Rakshit, "A decision support framework for airline flight cancellations and delays," *Transportation Science*, vol. 27, no. 3, pp. 266–280, 1993.

S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

D. Teodorovic and S. Guberinic, "Optimal dispatching strategy on an airline network after a schedule perturbation," *European Journal of Operational Research*, vol. 15, no. 2, pp. 178–182, 1984.

D. Teodorovic and G. Stojkovic, "Model for operational daily airline scheduling," *Transportation Planning and Technology*, vol. 14, no. 4, pp. 273–285, 1990.

A. Vasquez-Marquez, "American airlines arrival slot allocation system (ASAS)," *Interfaces*, vol. 21, no. 1, pp. 42–61, 1991.

G. Yu, "An Optimization Model for Airlines' Irregular Operations Control," in *Proc. of the Int. Symp. on Optimization Applications in Management and Engineering*, Beijing, 1995.

G. Yu, "Real-Time, Mission-Critical Decision Support Systems for Managing and Controlling Airlines' Operations," in *Proc. of the Int. Conf. on Management Science and Economic Development*, Hong Kong, 1996.

G. Yu and S. Luo, "Airline schedule perturbation problem caused by the ground delay program offered by FAA," *Transportation Science*, Forthcoming, 1997.