Available online at www.sciencedirect.com

ScienceDirect

Transportation Research Part B 42 (2008) 113–134

TRANSPORTATION RESEARCH PART B

www.elsevier.com/locate/trb

Reallocating arrival slots during a ground delay program

Jonathan F. Bard [a,*], Dinesh Natarajan Mohan [b,1]

[a] *Graduate Program in Operations Research and Industrial Engineering, 1 University Station, C2200, The University of Texas, Austin, TX 78712-0292, United States*

[b] *National Instruments, 11500 N. MoPac Expressway, Austin, TX 78759, United States*

Received 23 October 2006; received in revised form 2 July 2007; accepted 2 July 2007

Abstract

This paper presents a new model and solution methodology for the arrival slot reallocation problem faced by airlines when responding to a ground delay program (GDP). The objective is to reassign the flights in the GDP to time slots made available by the Federal Aviation Administration (FAA) such that flight delay and passenger missed connection costs are minimized. The problem is formulated as a dynamic program and solved with the help of branch and bound. Using data provided by American Airlines, initial tests showed that while the results were good for relatively small instances, as more flights were included, computation times grew exponentially. Given that the problem needs to be solved quickly in practice, the methodology was incorporated in a rolling horizon framework where larger problems are split into smaller subproblems and solved sequentially. This led to some degradation in solution quality but there was still considerable cost savings compared to the initial slot assignments proposed by the FAA. Computational experiments with both real and randomly generated data confirmed that problems of practical size could be solved within 5 min.

Keywords: Ground delay program; Dynamic programming; Rolling horizon; Branch and bound

1. Introduction

An imbalance between the arrival rate and the capacity of an airport can produce havoc in the flight schedule of an airline. Whenever visibility is reduced by bad weather, for example, arriving aircraft must be separated by wider than usual margins to ensure safety. This in turn reduces the flow through the airport. The Air Traffic Control System Command Center (ATCSCC) is the arm of the Federal Aviation Administration (FAA) that is responsible for monitoring weather conditions and keeping track of arrivals and capacity at

* Corresponding author. Tel.: +1 512 471 3076; fax: +1 512 232 1489.
E-mail addresses: jbard@mail.utexas.edu (J.F. Bard), Dinesh.Natarajan.Mohan@ni.com (D.N. Mohan).
[1] Tel.: +1 512 683 5342.

0191-2615/$ - see front matter
doi:10.1016/j.trb.2007.07.005

all airports in the United States. To match arrival rates to capacity, the ATCSCC has the authority to delay the take-off of flights scheduled to arrive at the target airport. When conditions warrant, a *ground delay program* (GDP) is issued. Delaying a flight on the ground is safer and less expensive than diverting or delaying it once airborne.

The purpose of this paper is to develop a model and solution methodology that can be used by an airline to assign flights to arrival slots with the objective of minimizing the costs incurred during a GDP. The problem is formulated as a dynamic program and ultimately solved within a rolling horizon framework due to the exponential growth of the state space. To further speed up the computations, a branch and bound procedure was embedded in the dynamic program along with a feasibility heuristic.

In 1999, 705 GDPs were issued for 23 airports resulting in 32.8 million minutes of total delay at an average of 90 min per controlled flight (Ryan, 2001). According to the Air Transport Association estimates, every minute of delay costs $29 to the airline (Chang et al., 2001; more recent statistics can be found at http://cdm.fly.faa.gov/ad/gdp.html). With about 600–1000 GDPs issued every year, efficient utilization of the arrival slots allocated to an airline can lead to tremendous savings. The question, then, is how to make the allocations in an optimal manner.

From the airlines' point of view, many factors must be considered when rescheduling flights during a GDP because delays propagate quickly in the network. All resources, including aircraft, cockpit crew, and cabin crew are shared between flights, so a GDP issued for a particular airport is likely to affect down-line operations. Standard regulations require that crew members be given a minimum rest period between flights and that they not work more than a given number of hours during a rotation. Moreover, each aircraft must undergo periodic maintenance checks, which can only be performed at a limited number of bases. If the schedule is disrupted for more than a few hours, it may not be possible to satisfy this requirement.

When responding to a GDP as well as to any other disruption, care must be taken to minimize the deviations from the original schedule. With respect to a GDP, airlines incur several types of costs associated with fuel and crew, missed passenger and crew connections, and the loss of customer good will. Costs due to flight cancellations consist of foregone revenues and per diem charges for food and lodging for the delayed passengers and crew.

Because weather is unpredictable, the ATCSCC may change the terms of the GDP midway into the program. When this happens, the airlines must be notified immediately. At the same time, any changes that the airlines make to their flight schedules must be reported to the ATCSCC as soon as they are announced. With all the conflicting objectives, the rescheduling process is both difficult and time-sensitive.

In the next section, we present a description of the GDP and the system currently in use. This is followed in Section 3 by a review of the literature and in Section 4 by a precise definition of the problem. The dynamic programming model is given in Section 5 and the full solution methodology is described in Section 6. To illustrate the computations, an example is given in Section 7, followed by our test results in Section 8. In Section 9, we draw some conclusions from the analysis.

2. Overview of the ground delay program

In 1981, the FAA initiated the GDP to tackle reduced ground capacity at airports affected by bad weather. The GDP contains the new arrival and departure times of the flights scheduled to arrive at the target airport and generally results in a combination of flight and passenger delays, flight cancellations, missed connections, delayed or missed aircraft maintenance, violation of crew legalities, and displacement of aircraft from the scheduled destinations. Each airline in a GDP must accept the new arrival and departure times (slots), but within some limits can rearrange the schedules of the affected flights. For example, if flights F1 and F2 are scheduled to arrive at 10:05 and 10:15, respectively, in the GDP, then if F1 and F2 originate from different airports, it is possible for the airline to interchange their arrival times.

Fig. 1 illustrates the effect of a GDP on flight schedules. In the figure, there are a total of 14 flights arriving from five different airports; arrows of similar style represent the flights arriving from the same airport. The FAA requires that the sequence of flights arriving from the same airport be maintained in the GDP. The original scheduled sequence and the period of bad weather conditions are shown on the timeline in the figure. The period of lower demand that follows the bad weather conditions can be used to absorb delayed flights.

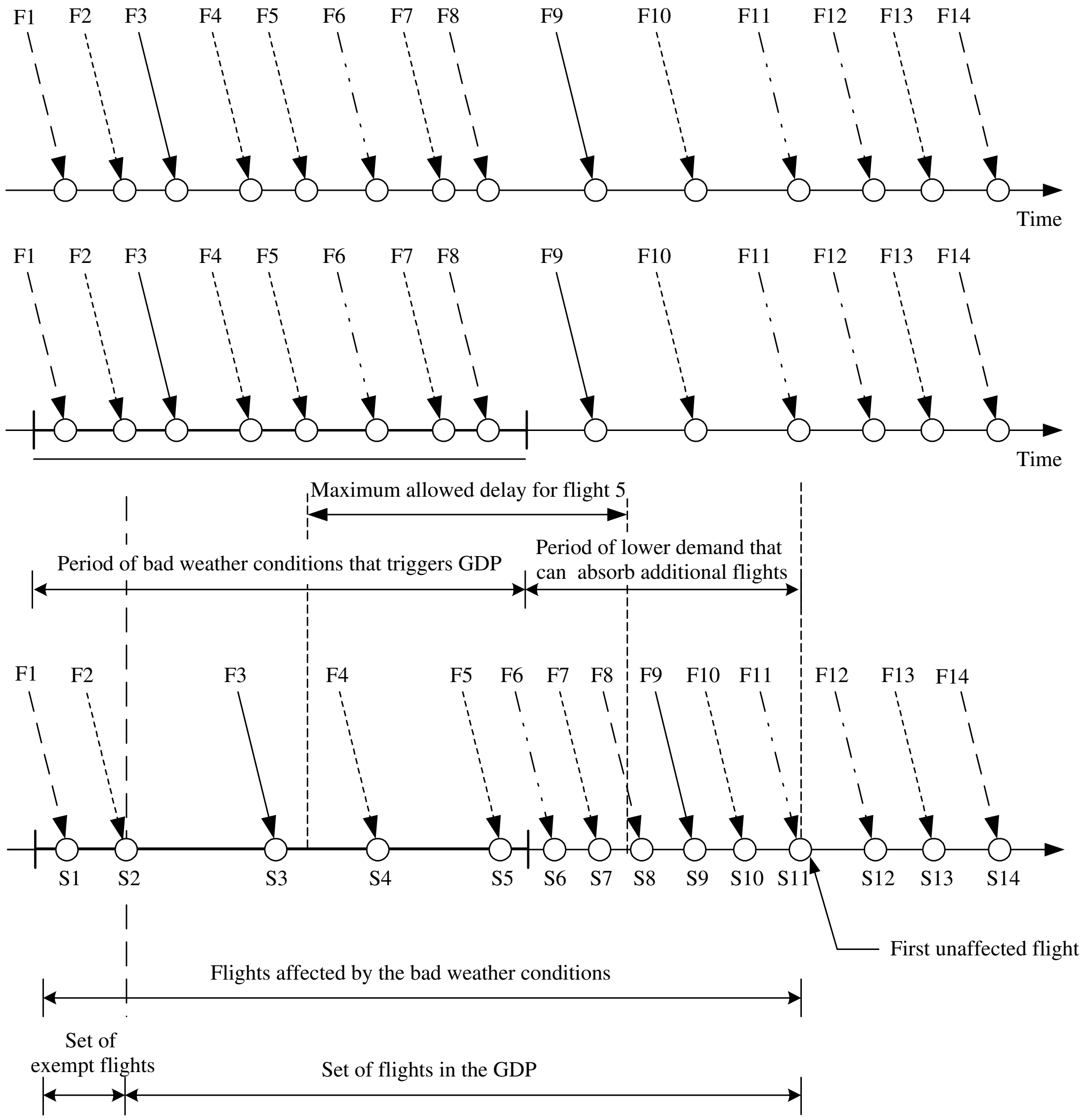

Fig. 1. Effect of a ground delay program on flight schedules.

In Fig. 1, S1 through S14 represent the arrival slots. Flights 11 through 14 are not affected by the GDP so they can arrive according to schedule. As the airport capacity is reduced during bad weather conditions, arrival slots originally scheduled during that period are pushed to the following period of lower demand. In the example, flights F1 and F2 are exempt from the GDP, which has been issued to flights F3 through F10.

Because of weaknesses in the design of the original GDP, the FAA has since developed a more comprehensive approach called Collaborative Decision Making (CDM), a joint government-industry initiative to improve air traffic flow management by better information sharing and new decision support mechanisms. Initial efforts began in 1991 when the FAA commissioned an analysis of the existing flight substitution process in an attempt to improve it. In 1993, it initiated an experiment called FAA Airline Data Exchange (FADE) to analyze whether the updated information that the airlines were providing could improve decision making. The test yielded positive results, which encouraged the continuation of FADE and further research activities. Additional testing proved that better information would result in reduced delays and sometimes even elimination of GDPs.

In the spring of 1995, CDM became official and the responsibilities of both the FAA and the airlines were formally defined. The role of the FAA is to monitor the national airspace and to keep the airlines informed of each airport's capacity, demand, weather and other conditions, while the airlines' role is to share with the

FAA its updated flight schedules, its demand, and its response to GDPs. The FAA is responsible for allocating arrival slots to the airlines but it is up to the airline to use those slots to the best of their advantage.

3. Literature review

Teodorovic (1985) was one of the first to investigate the disruptions to an airline's schedule that result from sudden changes in weather. He developed a network model in which the nodes represented flights and arcs specified flight precedence relations. The objective was to minimize the number of aircraft needed to serve the scheduled traffic. When more than one solution had the same minimum number of aircraft, the one with the least number of passengers affected by flight cancellations due to meteorological conditions was chosen. A simple heuristic using dynamic programming (DP) was provided to solve the model.

The ground holding problem (GHP), which is concerned with assigning optimal ground delays to flights to minimize costs incurred by an airline when landing capacities of the destination airports are reduced, is closely related to our work. The GHP is classified as static or dynamic depending on whether the ground delays assigned are updated with time as new and more accurate information becomes available.

Andreatta and Romanin-Jacur (1987) presented a GHP model to optimize total expected ground delay and airborne delay costs for a simplified single-airport problem. The aircraft scheduled to arrive at an airport were ordered according to the landing priorities, which may or may not be known a priori. The probabilities associated with the capacity of the airport were assumed to be stationary and known. The model was solved with a polynomial DP algorithm.

Vasquez-Marquez (1991) developed an interactive decision support tool for American Airlines to reduce ground delays. At the center of the methodology was an arrival slot allocation model that makes use of flight cancellations, mostly due to mechanical failures, to reduce the delays caused by a GDP. Whenever a flight in a GDP is cancelled due to mechanical problems, its slot becomes available for some other flight in the GDP. This leads to a new empty slot that can be filled by some other flight, and so on. In the paper, the author presented a traveling salesman problem formulation for the problem, where a salesman visits aircraft and delivers arrival slots. However, it appears that only an assignment problem is being solved with the objective of minimizing the total delay minutes with different weights given to different flight segments.

Terrab and Odoni (1993) addressed the GHP under deterministic and stochastic airport capacities separately. The problem with deterministic capacity was modeled as a minimum cost flow network with the objective of determining optimal ground delays assigned to aircraft to minimize the total delay costs subject to capacity constraints. Airborne delay costs were not considered because it is far less expensive to hold aircraft on the ground than in the air; also the airport capacity was assumed to be known. Their Fast Algorithm determines the landing sequence of aircraft for each time period based on the marginal delay costs. Landing priority is given to aircraft with higher marginal delay costs. The authors also discussed the stochastic version of the problem for both single-time period and multiple-time period cases. The exact DP approach presented for the multiple-time period case was an extension of the work by Andreatta and Romanin-Jacur (1987) for the single period problem. Because the approach was found to be extremely time and memory intensive, the authors provided four heuristics. They also showed that taking into consideration uncertainty in capacities results in significant savings in delays.

Richetta and Odoni (1993, 1994) formulated the static probabilistic GHP as a stochastic linear program and its dynamic version, respectively. Airborne delays were allowed in both the models. More recently, Rossi and Smriglio (2001) modeled the GHP as a set packing problem and developed a branch-and-cut algorithm to solve it.

Vranas et al. (1994a) were the first to address the static, deterministic version of the GHP for a network of airports and considered the down-line delay effects. They presented three integer programming (IP) models with the first considering only ground delays, the second considering both ground and airborne delays, and the third incorporating flight cancellations. In all cases, only single connections were considered; i.e., a flight could have no more than one successive flight and no more than one preceding flight.

Vranas et al. (1994b) also addressed the dynamic multi-airport GHP and presented several IP models that updated the ground and airborne delays when new and more accurate weather forecasts became available.

Both deterministic and probabilistic versions of the problem were discussed along with many extensions that considered flight cancellations, en route speeds, interdependent arrival and departure capacities, and banking.

Mehndiratta and Keifer (2003) presented predictions of delay and service impacts of slot controls as a demand management tool for the San Francisco International Airport in 2015. The slot allocation mechanism employed in the study was market based, where the flights were evaluated according to the revenue they were expected to generate. The authors discussed the estimated impacts on delays, demand management and passenger service.

In work somewhat related to ours, Luo and Yu (1997) presented an IP formulation to minimize the total number of flights delayed at an airport due to a GDP. Their model was equivalent to an assignment problem with side constraints in which flights were assigned to arrival slots to minimize total incoming and outgoing delays. The authors proved that the general problem is NP-hard and proposed a polynomial time heuristic to solve a special case which makes the following assumptions: (1) At most two resources are needed for flights, (2) There is no arrival slot between departure times of two outbound flights that need a resource from a particular inbound flight, and (3) All flight-slot assignments are feasible. The heuristic guarantees a solution that is within one from the optimum. The authors then derived valid inequalities for the IP to strengthen the LP relaxation.

In a follow-on paper, Luo and Yu (1998) discussed several techniques to tackle perturbations due to GDPs. They proposed a number of models that assign flights to landing slots to minimize either canceled flights, delayed flights, the maximum delay, passenger delays, or total delay costs. For work on irregular operations related to more general scheduling disturbances than the GDP, see Bard et al. (2001), Rakshit et al. (1996), Stojkovic et al. (2002), Teodorovic and Stojkovic (1995), Thengvall et al. (2001), Yan and Lin (1997), Yan and Yang (1996), Yan and Young (1996), Yan and Tu (1997).

4. Problem description

When the FAA assigns flights to arrival slots during GDPs, it is the responsibility of the airline to use the slots in the most efficient manner. The problem under consideration is the reallocation of the flights to arrival slots such that some objective is optimized. Possible objectives include minimizing passenger delay minutes, minimizing the number of delayed flights, minimizing flight cancellations, and optimizing the FAA dependability statistic.

The model proposed in this paper addresses the static, deterministic version of the single-airport GHP. The objective chosen is to assign the arriving flights in the GDP to the available arrival time slots so that total cost, as measured by the dollar value of passenger, aircraft and crew delays, plus compensation to passengers who miss their connection flights, is minimized. Original flight schedules, data concerning the number of passengers traveling and connecting between flights, and arrival time slots available to the airline form the input to the model. The output is the new allocation of slots to flights. Although flight cancellations could be readily accommodated in our approach, we chose not to include them because an airline will only cancel a flight as a last resort. GDPs do not require cancellations. From the point of view of each affected airline, flights of other airlines in the GDP are assumed to arrive and depart as scheduled and are not considered to be part of the GDP in the model.

The fact that weather is the primary trigger for a GDP implies a degree of uncertainty in the planning process. In anticipation of a GDP of a certain length, an airline might develop a set of scenarios to minimize the potential disruption to its schedule, implementing the one that is most cost-effective for the realized situation. Nevertheless, once the GDP is issued, there is a question of whether it will remain in effect for the stated time period or whether an updated program will be issued as conditions change. Dealing with these uncertainties is an important topic for research but beyond the scope of our work.

5. Model formulation

When assigning a flight to a time slot we will consider the following three costs: (1) passenger and crew delays, (2) additional fuel consumption, and (3) compensation of passengers who miss their connections. For passengers who are connecting to outbound flights whose inbound flights are not in the GDP, the third cost component can be evaluated easily when the departing times of the outbound flights are known. For passengers who are connecting to outbound flights whose inbound flights are in the GDP, however, the missed connection costs can be computed only if the slot assignments of those inbound flights are known. Thus, the cost of assigning a flight to a slot depends on the other assignments and so cannot be specified a priori. This precludes the use of standard assignment-type optimization models unless a quadratic objective function is considered. Because the state-of-the-art in solving quadratic assignment problems is limited roughly to instances no larger than 25×25, we take a different approach (Erdoğan and Tansel, 2007).

5.1. Assumptions

In order to develop an approximate backwards DP formulation of the GDP, four assumptions are necessary. Suppose there are n inbound flights in a GDP and n arrival slots. Let t_m indicate the starting time of slot m, where $m = 1, \ldots, n$ with $t_1 < t_2 < \cdots < t_n$, and consider all passengers arriving on inbound flights in the GDP connecting to flights whose inbound flights are also in the GDP. Let $\phi(j)$ be an outbound flight in the GDP whose corresponding inbound flight j is also in the GDP.

In constructing the DP, consider the nth arrival slot only. The first two cost components of assigning any flight to this slot are straightforward to compute since the arrival time of the flight will be t_n and the scheduled flight arrival times and the number of passengers and crew members traveling are known. To address the passenger missed connection costs, we introduce the following.

Assumption 1. A passenger on arriving flight i assigned to slot m in the GDP and connecting to outbound flight $\phi(j)$ whose corresponding inbound flight j is also in the GDP and is assigned slot l, will miss his connection if $t_l < t_m$ and $d_{\phi(j)} < t_m$, where $d_{\phi(j)}$ is the scheduled departure time of $\phi(j)$.

Assumption 2. The departure capacity of the GDP airport is not restricted, i.e., flight $\phi(j)$ in the GDP can depart at max $\{t_l + \Delta, d_{\phi(j)}\}$, where t_l is the arrival time of flight j (which is assigned slot l) and Δ is the minimum turnaround time needed for aircraft between flights.

Assumption 3. A passenger on arriving flight i assigned to slot m in the GDP and connecting to an outbound flight $\phi(j)$ whose corresponding inbound flight j is also in the GDP and is assigned slot l, will make his connection if $t_l > t_m$.

Assumption 4. All flights not in the GDP arrive and depart according to the original schedule.

Because all other inbound flights in the GDP must be assigned to prior slots, the missed connection costs of passengers who are connecting to outbound flights whose predecessors are in the GDP can be computed due to Assumptions 1 and 2. The missing connection costs of passengers who are connecting to outbound flights whose predecessors are not in the GDP can be computed due to Assumption 4. Now let us suppose that flight i is assigned to slot n. Then the cost of assigning any other flight j to slot $n - 1$ can be calculated similarly by Assumptions 1, 2 and 4. Due to Assumption 3, passengers on flight j who connect to flight $\phi(i)$, outbound flight of i, will not miss their connection.

While Assumptions 1–4 have a practical justification, they may be violated in some situations. In particular, passengers who were assumed to have missed their connections as a result of Assumption 1 might not actually miss them. For example, let us consider two flights i and j in the GDP that are assigned to arrival time slots 09:08 and 09:10, respectively, and suppose that the scheduled departure times of their outbound flights are 08:45 and 08:50. If the minimum turnaround time is 30 min, then the aircraft associated with flight i will be ready to depart at 09:38 and passengers who arrive on flight j and are scheduled to connect to $\phi(i)$, the outbound flight of i, will make their connections if the passenger turnaround time happens to be less than 28 min. But according to Assumption 1, the airline would incur missing connection costs.

Similarly, passengers who were assumed to have made their connecting flights as a result of Assumption 3 might not really make their connections, especially when the arrival time slots are close together. The failure of Assumption 3, however, is likely to be negligible. Nevertheless, the solution provided by the dynamic program constructed under these assumptions may overestimate the total costs and so may not be optimal. This issue is addressed in the solution procedure explained in Section 6.

5.2. Dynamic programming model

Based on backward recursion, the arrival slot reallocation problem is now formulated as a DP. The following notation is used in the developments:

Indices and sets

N	set of flights in the GDP; $n = \|N\|$
N_i	set of flights in the GDP excluding flight i; $N_i = \{1, \ldots, i-1, i+1, \ldots, n\}$
S	subset of N_i; $S \subseteq N_i$
k	cardinality of S; $k = \|S\|$
R	set of outbound flights whose predecessors are in the GDP; $r \in R$
O	set of outbound flights whose predecessors are not in the GDP; $o \in O$
$\sigma(S)$	set of outbound flights whose predecessors are in set S

Parameters

$\theta_{ij}(S)$	cost of assigning flight $j \in S$ to slot $n-k+1$ when flight i occupies slot $n-k$ and all the remaining flights in S have previously been assigned to the last $k-1$ slots
$\delta(j)$	origin airport of flight j; $j = 1, \ldots, n$
a_j	originally scheduled arrival time of flight j; $j = 1, \ldots, n$
d_j	originally scheduled departure time of flight j; $j \in R \cup O$
t_m	starting time of arrival slot m; $m = 1, \ldots, n$
Δ	minimum turnaround time for an aircraft between two flights
τ	minimum turnaround time for passengers connecting between flights
D	maximum delay allowed for a flight
c_{i1}	cost of assigning flight i to the first arrival slot
P_j	cost of one minute of delay for passengers on flight j
W_j	cost of one minute of delay for crew on flight j
L_j	cost of fuel for one minute of delay of flight j
M_{jr}	cost incurred by the airline in compensating passengers on flight j when they miss their connection flight r (function of the number of connecting passengers)

Variables

V_{jik}	indicator variable that identifies whether passengers arriving on inbound flight j miss the outbound flight $\phi(i)$ associated with inbound flight i, with flights i and j occupying slots $n-k$ and $n-k+1$, respectively; equals 1 if passengers on flight j do not miss flight $\phi(i)$ (that is, $t_{n-k} + \Delta \geqslant t_{n-k+1} + \tau$), 0 otherwise

Optimal value function

$$f_k(S, i) = \text{minimum cost of assigning the last} k \text{slots to arriving flights in } S \text{ when flight } i \text{ occupies slot } n-k \tag{1a}$$

Recurrence relation

$$f_k(S, i) = \min_{j \in S}\{f_{k-1}(S \setminus \{j\}, j) + \theta_{ij}(S)\}, \quad k = 1, \ldots, n-1, \;\; i = 1, \ldots, n, \;\; S \subseteq N_i \tag{1b}$$

such that $|S| = k$ and where the cost function $\theta_{ij}(S)$ is given by

$$\theta_{ij}(S) = (P_j + W_j + L_j)(t_{n-k+1} - a_j) + \sum_{\substack{o \in O \\ d_o \leqslant t_{n-k+1} + \tau}} M_{jo} + \sum_{\substack{r \in R \setminus \sigma(S) \\ d_o \leqslant t_{n-k+1} + \tau}} M_{jr} - M_{j,\phi(i)} V_{jik} \tag{1c}$$

and state (S,i) is restricted as follows:

$$a_i \geqslant a_j \quad \text{for all } j \in N \setminus S \text{ when } \delta(i) = \delta(j) \tag{1d}$$

$$0 \leqslant t_{n-k} - a_i \leqslant D \tag{1e}$$

$$\textit{Boundary conditions}: \; f_0(-,i) = 0, \quad i = 1,\ldots,n \tag{1f}$$

$$\textit{Solution}: \; \min_{i \in N}\{f_{n-1}(N_i,i) + c_{i1}\} \tag{1g}$$

The cost of assigning flights in S to the last k slots when flight i occupies slot $n-k$ can be found by determining the cost of assigning the flights other than, say j, to the last $k-1$ slots when flight j occupies slot $n-k+1$, and adding the cost of assigning j to that slot when flight i occupies slot $n-k$. The minimum cost of assigning the flights in set S to the last k slots when flight i occupies slot $n-k$ can be found by considering all $j \in S$. This is represented by the recurrence relation (1b).

The function in (1c) represents the cost of assigning flight j to slot $n-k+1$ when the rest of the flights in set S are assigned optimally to the last $k-1$ slots and flight i occupies slot $n-k$. It is the sum of the costs directly associated with the delay and the missed connections. The first term corresponds to the delay minutes incurred by assigning flight j to slot $n-k+1$. The second term represents the missing connection costs incurred for those passengers who connect to flights whose inbound flights are not in the GDP. This can be calculated based on Assumption 4. The third term represents the missing connection costs incurred for those passengers who connect to flights whose inbound flights are in the GDP. This is computed based on Assumption 1. The costs associated with passengers on flight j who miss flight $\phi(i)$ are also included in this term. Because flight i occupies slot $n-k$, it is possible to determine whether or not passengers arriving on flight j will actually miss flight $\phi(i)$. This is taken into account with the variable V_{jik}. As indicated by the fourth term, the corresponding cost is subtracted from the total cost if flight j does not miss flight $\phi(i)$.

Constraint (1d) is included to ensure that all flights in the GDP that originate from the same airport are assigned to slots in the same order as their original arrival times. For instance, if flights x and y depart from the same airport and x was originally scheduled to arrive before y at the GDP airport, then in the new slot allocation, x must arrive before y. Constraint (1e) ensures that the arrival time assigned to flight i in the GDP, t_{n-k}, does not precede its scheduled arrival time, a_i, and that the delays assigned to flights are restricted to a maximum value D. When the set S is empty, the optimal value function is zero. This is represented by Eq. (1f) as a boundary condition. Because the function $f_k(S,i)$ does not include the cost of assigning flight i to slot $n-k$, it needs to be added to represent the total cost incurred. When $k = n-1$, Eq. (1g) is used to calculate the optimal cost of assigning the n flights to the n arrival slots.

In the worst case, it is necessary to evaluate the cost function $f_k(S,i)$ in (1b) for all $O(n2^n)$ states. At each state (S,i), the function $\theta_{ij}(S)$ must be evaluated for each $j \in S$, which requires $O(n^2)$ computations. In total then, the DP has time complexity $O(n^3 2^n)$. Of course, the need to preserve the order of flights originating from the same airport would make some states infeasible and hence reduce the computational burden.

6. Solution procedure

A state in the DP is represented by (S,i), where set S is a subset of N_i, which is the set of all flights in GDP excluding flight i. Because the slot reallocation problem needs to be solved in near real time to enable the flight controllers to make quick decisions, exhaustively enumerating the state space and evaluating Eq. (1b) is not likely to be efficient. To improve performance, branch and bound will be applied within the DP to eliminate states that cannot lead to the optimal solution. This will reduce overall storage requirements and speed convergence. The solution procedure is presented below.

The slot assignments given by the ATCSCC when it issues the GDP provide an initial feasible solution to the slot reallocation problem. The cost associated with the solution serves as an initial upper bound (UB), which we then improve as much as possible with a pairwise exchange procedure. In this procedure all pairs of arriving flights i and j and their corresponding arrival slots are interchanged and the cost of the new sequence computed. If a reduction results, the exchange is made. The process is repeated for all pairs until a local solution is reached.

The DP computations start at stage 1 where the initial states $(-,i)$ give values of zero for $f_0(-,i)$, $i = 1,\ldots,n$. There are $O(n-1)$ states at stage 1, i.e., $(\{i\},j)$, $j \neq i$. Let S_k and i_k be the set S and flight i in stage k. Let N_{ik} be the set of flights excluding flight i_k. State (S_{k+1}, i_{k+1}) can be generated from state (S_k, i_k) as follows:

$$S_{k+1} = S_k \cup \{i_k\} \tag{2}$$

$$i_{k+1} \in N_{ik} \setminus S_k \tag{3}$$

As mentioned, a slight error in the cost calculation may occur because the passengers arriving on flights in set S who were assumed to miss certain connecting flights may not really miss them. Nevertheless, as more flights are assigned to slots and the procedure moves from one stage to the next, the cost $f_k(S,i)$ can be updated to compensate for any error. For example, let us consider state (S,i) where flight i occupies slot $n-k$ and let us assume that $t_{n-k} + \Delta > d_{\phi(i)}$. Hence flight i will depart by time $t_{n-k} + \Delta$ and passengers on the inbound flights in the GDP that are assigned to slots between t_{n-k} and $t_{n-k} + \Delta - \tau$ will not miss flight i. But according to Assumption 1, some passengers on those inbound flights will incur missed connection costs with flight i. A portion of the corresponding error in the calculation of $f_k(S,i)$ can be corrected, though, by updating those costs after it is known that flight i will occupy slot $n-k$. This will provide a more accurate estimate of the actual costs, but the true optimum may still be overlooked if it was fathomed at an earlier stage.

Feasible states in any stage are generated only from unfathomed states in the previous stage. The fathoming criterion is given by

$$f_k(S,i) + q(S,i) + C(S,i) \geqslant \mathrm{UB} \tag{4}$$

where $f_k(S,i)$ is the minimum cost of assigning the flights in set S to the last k slots when flight i occupies slot $n-k$; $q(S,i)$ is a lower bound for a partial schedule to state (S,i) and can be calculated by solving the assignment problem (5) given below; and $C(S,i)$ is the cost of assigning flight i to slot $n-k$ given the solution to (5) and the assignments associated with $f_k(S,i)$.

Assignment problem

Indices

j index for arriving flights in the GDP, $j \in N_i \backslash S$

m index for remaining arrival time slots; $m = 1,\ldots,n-k-1$

Parameters

c_{jm} delay costs plus missing connection costs associated with outbound flights whose predecessors are not in the GDP or in $S \cup \{i\}$ that are incurred by assigning arriving flight $j \in N_i \backslash S$ to slot $m \in \{1,\ldots,n-k-1\}$ for current state (S,i)

Decision variables

x_{jm} (binary) 1 if arriving flight j is assigned to slot m; 0 otherwise

The coefficients c_{jm} must be calculated for each state (S,i). As in (1c), the delay costs are equal to $(P_j + W_j + L_j)(t_m - a_j)$. Missed connection costs are included only for passengers connecting to flights not in the GDP because those arrivals and departures are known. Suppose flights α and β are not in the GDP and that flights r and j are. Looking at Fig. 2 where flight j is assigned to slot m, passengers connecting to outbound flight $\phi(\alpha)$ miss their connections. This cost is known and so is included in c_{jm}. With respect to arriving flight β, passengers on flight j connecting to outbound flight $\phi(\beta)$ will make their connection so the corresponding missed connection costs are zero. Similarly, the missed connection costs for passengers connecting to outbound flights associated with the flights in the set $S \cup \{i\}$ are also zero because the latter flights arrive after flight j.

What is not included in c_{jm}, and hence what makes it an underestimate, is the missed connection costs associated with passengers arriving on flight $j \in N_i \backslash S$ and connecting to outbound flight $\phi(r)$, where flight r is in the GDP but not in $S \cup \{i\}$. In Fig. 2, flight r is assigned to slot l and arrives before flight j which is assigned to

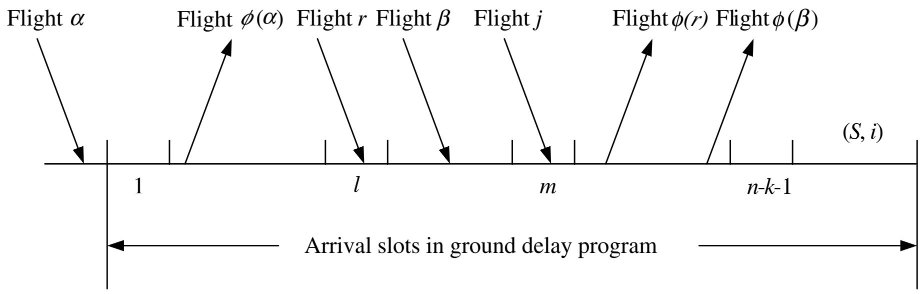

Fig. 2. Example of arriving and departing flights not included in state (S, i).

slot m. Because the departure time of flight $\phi(r)$ is later than the arrival time of flight j, passengers arriving on j and connecting to $\phi(r)$ will miss their flight and incur a missed connection cost if $t_j + \Delta - \tau > d_{\phi(r)}$. However, we don't know the respective arrival times of flights r and j so we cannot include the associated missed connection costs, should there be any, in c_{jm}.

Model for S, i and k fixed

$$\text{Minimize} \quad \sum_{j \in N_i \setminus S} \sum_{m=1}^{n-k-1} c_{jm} x_{jm} \tag{5a}$$

$$\text{subject to} \quad \sum_{j \in N_i \setminus S} x_{jm} = 1, \quad m = 1, \ldots, n-k-1 \tag{5b}$$

$$\sum_{t=1}^{n-k-1} x_{jm} = 1, \quad j \in N_i \setminus S \tag{5c}$$

$$x_{jm} \in \{0, 1\}, \quad m = 1, \ldots, n-k-1; \; j \in N_i \setminus S \tag{5d}$$

The objective in (5a) is to minimize the total costs incurred by assigning the flights that are not part of the state (S, i) to the remaining time slots $m = 1, \ldots, n - k - 1$. In the calculation of c_{jm}, it is assumed, as stated above, that passengers arriving on flight j will make their connection if their outbound flight is $\phi(r)$, for all $r \in N_i \backslash S$ and $r \neq j$. Constraint (5b) ensures that a time slot is assigned to only one flight while (5c) guarantees that a flight is assigned to only one time slot. Constraint (5d) guarantees integrality. In (5a), the coefficients c_{jm} do not include the missing connection costs with outbound flights whose predecessors are in GDP because they cannot be computed unless all the assignments are known.

Although not pursued here, a second, and perhaps better lower bound may be obtained by formulating the partial problem associated with (5) more fully as a quadratic program (QP). Letting $\overline{N} = N_i \setminus S$, the objective function in such a formulation would be

$$\sum_{j \in \overline{N}} \sum_{m \in \overline{N}} \sum_{\substack{k \in \overline{N} \\ k < j}} \sum_{l \in \overline{N} \setminus \{m\}} c^{\mathrm{PI}}_{jmkl} x_{jm} x_{kl} + \sum_{j \in \overline{N}} \sum_{m \in \overline{N}} \left(c^{\mathrm{D}}_{jm} + c^{\mathrm{PO}}_{jm} \right) x_{jm}$$

where

c^{PI}_{jmkl} misconnection cost for passengers arriving on flight j and connecting to flight k in the GDP when flight j is assigned to slot m and flight k is assigned to slot l (because aircraft turnaround time is at least as great as passenger connect time, c^{PI}_{jmkl} will only be nonzero for $j > k$)

c^{D}_{jm} delay cost when flight j is assigned to slot m

c^{PO}_{jm} misconnection cost for passengers arriving on flight j and connecting to flights that are not in the GDP when flight j is assigned to slot m

For $|\overline{N}| \leqslant 15$, it should be possible to solve the QP to optimality in a matter of minutes. However, because it is desirable to compute a lower bound $q(S, i)$ for each state (S, i), this is far too time consuming, implying that the only practical way to proceed would be to solve a relaxation of the QP. Recent advances in bound computations for the quadratic knapsack problem (e.g., see Pisinger et al., 2007) offer a good start pointing for the computations needed here.

In any case, the optimal solution to (5) provides a lower bound on the cost of a partial feasible solution on the path to state (S,i). When this cost is added to the updated value of $f_k(S,i)$, given $C(S,i)$ and the assignments from (5), it gives the least cost possible for any solution which contains state (S,i) at stage k. When this value is greater than the current upper bound UB, (S,i) is fathomed since it is not possible to find a better solution by pursuing this state further. If (S,i) cannot be fathomed, then an upper bound, call it $u(S,i)$, is computed. This value is equal to the cost associated with any feasible set of slot assignments that contains (S,i) at stage k. We use the slot assignments given by $f_k(S,i)$ and the assignment of the remaining flights to slots 1 through $n-k-1$ in the order of their scheduled arrival times, with flight i occupying slot $n-k$. If this assignment is infeasible, then there is no feasible solution to state (S,i) and it can be fathomed. If feasible, pairwise slot exchanges are performed to improve the cost, and $u(S,i)$ is updated. If the updated value of $u(S,i)$ is less than the current upper bound, it will become the new upper bound when assessing further states.

The procedure is repeated until stage $n-1$ is reached or when there are no more feasible states. At this point, the current upper bound UB represents the cost of the best available solution. Pairwise slot exchanges are then performed once again in an attempt to reach a local optimal solution. An outline of the computation flow follows.

Slot_Allocation_Dynamic_Programming_Algorithm

1. Input: FAA issued GDP
2. Initialization:
   - 2.1 Construct initial solution from input and perform pairwise slot exchanges to get UB. Store the corresponding slot assignments σ^*.
   - 2.2 Set $S=\emptyset$; generate all feasible states (S,i) for all $i=1,\ldots,n$.
   - 2.3 Set $\Psi=\emptyset$, where Ψ stores 'good' states for each stage.
3. For each (S,i),

```
{
Update f_{k-1}(S\{j},j) values, j ∈ S, given that flight i occupies slot n − k.
Compute f_k(S,i).
Compute q(S,i), lower bound for partial assignment of flights, by solving
assignment problem (5).
Compute C(S,i), cost of assigning flight i to slot n − k given the assignments
from (5) and the assignments associated with f_k(S, i).
Update f_k(S,i) given the assignments from the assignment problem.
If f_k(S,i) + q(S,i) + C(S,i) ⩾ UB,
   then fathom state (S,i);
Else
   Compute u(S,i), upper bound for a feasible set of slot assignments, given the assignments of f_k(S,i)
   and flight i occupying slot n − k.
   If infeasible
      then fathom state (S,i);
   Else
      improve u(S,i) by performing feasible pairwise exchanges
      If u(S,i) ⩽ UB,
         then set UB = u(S,i) and store the slot assignments in σ*.
      End if
      Add state (S,i) to Ψ; that is, put Ψ ← Ψ ∪ (S,i).
   End if
End if
}
```

4. If $\Psi = \emptyset$, then go to Step 7.
5. Generate all feasible states for next stage from all elements in Ψ using Eqs. (2) and (3).
6. Repeat Steps 3–5 until $|S| = n - 1$.
7. UB is the least cost achieved by the algorithm and σ^* represents the slot assignments associated with the least cost.

7. Illustrative example

A four-aircraft example with $\Delta = 30$ and $\tau = 20$ will be used to illustrate the application of the DP algorithm to the slot reallocation problem. Table 1 gives the schedule of the flights of an airline affected by a GDP at a particular airport. The third column lists the scheduled arrival times for the inbound flights A1, A2, A3, and A4, respectively, and the fourth column gives the corresponding arrival slots in the GDP. Outbound connecting flights and scheduled departure times are listed in the last two columns. Because all of the GDP slots are later than 08:50 + 0:30, the latest of the scheduled departure times of the outbound flights of the four-aircraft under consideration plus the turnaround time, all aircraft will depart 30 min after they arrive. To allow for maximum flexibility, it is assumed that each flight originates from a different airport, which means that the new arrival sequence can be completely different than the original.

Table 2 shows the delay in minutes incurred [denoted by $t_{n-k+1} - a_j$ in Eq. (1c)] when the inbound flights are assigned to the different time slots available. For instance, assigning inbound flight A2 to the 11:00 slot results in a 230-min delay. In the example, the cost of a minute of delay, accounting for crew, passengers and fuel, is assumed to be \$1 [denoted by $P_j + W_j + L_j$ in Eq. (1c)]. Table 3 presents the costs associated with missed connections with the outbound flights whose predecessors are not in the GDP [denoted by M_{jo} in Eq. (1c)]. If flight A3 is assigned to the 11:05 slot, for example, then the airline would incur a cost of \$150 to com-

Table 1
Schedule of flights under a GDP

Aircraft	Inbound flight	Scheduled arrival time	GDP arrival slot	Outbound flight	Scheduled departure time
1	A1	07:00	10:10	B1	08:10
2	A2	07:10	10:55	B2	08:25
3	A3	07:20	11:00	B3	08:45
4	A4	07:30	11:05	B4	08:50

Table 2
Delay minutes incurred, $t_{n-k+1} - a_j$

Inbound flights	Time slots			
	10:10	10:55	11:00	11:05
A1	190	235	240	245
A2	180	225	230	235
A3	170	215	220	225
A4	160	205	210	215

Table 3
Missed connection costs with outbound flights whose predecessors are not in the GDP, M_{jo}

Inbound flights	Time slots			
	10:10	10:55	11:00	11:05
A1	\$100	\$120	\$130	\$135
A2	\$75	\$85	\$85	\$95
A3	\$120	\$130	\$145	\$150
A4	\$50	\$75	\$80	\$90

pensate passengers who were connecting between A3 and outbound flights whose predecessors are not in the GDP.

Missed connection costs [denoted by M_{jr} in Eq. (1c)] for passengers arriving on a flight in the GDP who miss their connecting flights whose predecessors are also in the GDP are presented in Table 4. When flight A1 misses outbound flight B3, for example, the airline incurs a cost of $50. Passengers on flights that are assigned to slots 10:55, 11:00 and 11:05 connecting to the outbound flight whose predecessor is assigned to the 10:10 slot will miss their connections. In all other cases, however, the intervals between the time slots are such that passengers will make their connections.

The initial upper bound UB is the cost associated with the slot assignments given in the GDP; i.e., flights A1, A2, A3 and A4 assigned in that order to the four time slots. The total cost is $1385 – the sum of the delay costs and the two types of missed connection costs corresponding to those slot assignments. The boundary conditions of the DP, as given in Eq. (1f), are

$$f_0(-,i) = 0, \quad i \in \{A1, A2, A3, A4\} \tag{6}$$

For $|S| = 1$, states (S, i) at the first stage of the algorithm are generated by Eqs. (2) and (3). Table 5 lists the corresponding costs of assigning the flights in S to the 11:05 slot when flight i occupies the 11:00 slot. To see how $\theta_{ij}(S)$ is calculated, consider state 1 where $S = \{A1\}$, $i = A2$ and $j = A1$. There are four terms in Eq. (1c), which we denote by θ^1, θ^2, θ^3, θ^4. The cost of assigning flight A1 to slot 4 (11:05) when flight A2 occupies slot 3 (11:00) requires the calculation of each of these terms.

θ^1: In the example, the cost of a minute of delay, accounting for crew, passengers and fuel, is assumed to be $1. Hence, θ^1 equals the delay minutes of flight A1 when assigned to slot 4 (11:05). The delay minutes are provided in Table 2 and in this case are 245 (time from 07:00 to 11:05), so $\theta^1 = \$1/\text{min} \times 245\ \text{min} = \$245$.

θ^2: This term is $\sum_{\substack{o \in O \\ d_o \leqslant t_{n-k+1}+\tau}} M_{jo}$ and represents the missing connection costs incurred for those passengers who connect to flights whose inbound flights are not in the GDP. Hence they are known and provided in Table 3. Assigning A1 to slot 4 gives $\theta^2 = \$135$.

Table 4
Missed connection costs with outbound flights whose predecessors are in the GDP

Inbound flights	Outbound flights			
	B1	B2	B3	B4
A1	–	$35	$50	$20
A2	$40	–	$35	$30
A3	$60	$20	–	$50
A4	$15	$45	$25	–

Table 5
Stage 1 DP cost calculations for the 11:05 slot

State	S	I	Slot assignments	j	$\theta_{ij}(S)$	$f_{k-1}(S\backslash\{j\},j)$	$f_k(S,i)$
1	A1	A2	__ __ A2 A1	A1	450	0	450
2	A1	A3	__ __ A3 A1	A1	435	0	435
3	A1	A4	__ __ A4 A1	A1	465	0	465
4	A2	A1	__ __ A1 A2	A2	395	0	395
5	A2	A3	__ __ A3 A2	A2	400	0	400
6	A2	A4	__ __ A4 A2	A2	405	0	405
7	A3	A1	__ __ A1 A3	A3	445	0	445
8	A3	A2	__ __ A2 A3	A3	485	0	485
9	A3	A4	__ __ A4 A3	A3	455	0	455
10	A4	A1	__ __ A1 A4	A4	375	0	375
11	A4	A2	__ __ A2 A4	A4	345	0	345
12	A4	A3	__ __ A3 A4	A4	365	0	365

θ^3: This term is $\sum_{\substack{r\in R\setminus\sigma(S)\\ d_r\leqslant t_{n-k+1}+\tau}} M_{jr}$ and represents the missing connection costs incurred for those passengers who connect to flights whose inbound flights are in the GDP; that is, flights A2, A3 and A4. According to Assumption 1, A1 will miss B2, B3 and B4. Hence the total cost associated with A1 missing B2, B3 and B4 is $\theta^3 = \$35 + \$50 + \$20 = \$105$.

θ^4: The costs associated with passengers on flight A1 who miss flight B2 are included in the third term. Because we know that flight A2 occupies slot 3 (11:00), it is possible to determine whether or not passengers arriving on flight A1 will actually miss flight B2. Flight A2, assigned to slot 3, will depart at 11:30 as B2 (it arrives at 11:00 and requires 30 min for turnaround). Passengers on A1, which is assigned to slot 4 (11:05), will be ready to leave by 11:25 (passenger turnaround time $\tau = 20$ min). Because 11:25 < 11:30, passengers arriving on A1 and connecting to B2 will not miss their flight. Therefore, the corresponding missed connection costs, $M_{j,\phi(i)} = M_{\mathrm{A1},\phi(\mathrm{A2})} = M_{\mathrm{A1,B2}} = \$35$, that were included in θ^3 must be subtracted out. This means that $V_{jik} = V_{\mathrm{A2,A1},1} = 1$ in the fourth term and that $\theta^4 = \$35$.

From these calculations, we have

$$\theta_{\mathrm{A2,A1}}(\{\mathrm{A1}\}) = \theta^1 + \theta^2 + \theta^3 + \theta^4 = \$245 + \$135 + \$105 - \$35 = \$450$$

The next step in the procedure is to eliminate any states that cannot lead to the optimal solution. For each state, the assignment problem (5) is solved to find a lower bound, $q(S,i)$, on the cost of assigning the flights other than i and the one in S to the 10:10 and 10:55 slots. After obtaining the solution, the cost function $f_k(S,i)$ is updated. The cost $C(S,i)$ of assigning flight i to the 11:00 slot is also computed.

Table 6 gives the entire set of fathoming calculations for stage 1. Column 'LB' identifies the lower bound or least cost solution achievable to state (S,i); i.e., the sum of $f_k(S,i)$, $q(S,i)$ and $C(S,i)$. If it is found to be greater than or equal to the existing UB, then (S,i) is fathomed. In the example, states 3 and 10 are fathomed. Column $u(S,i)$, computed for all unfathomed states, gives the cost associated with a feasible set of slot assignments for all four flights. A feasible solution is obtained by assigning the flight in S to the 11:05 slot, flight i to the 11:00 slot and the two other flights to the remaining slots in the order in which they were originally scheduled to arrival. Pairwise slot exchanges are then performed in an attempt to reduce the cost. When the 'improved $u(S,i)$,' the last column in Table 6, is less than the existing UB, it becomes the new UB. The last column in Table 6 shows the value of UB after assessing the state.

If we look at state 1, for example, we see that the pairwise exchanges reduced $u(S,i)$ from \$1375 to \$1350, which is better than the current UB (\$1385), so \$1350 becomes the new UB with ordering A4 → A3 → A2 → A1. Note that without the pairwise exchanges, UB would have been \$1375 and state 3 would not have been fathomed. Similarly, state 10 would not have been fathomed.

States at the second stage ($k = 2$) are generated from the states that remain from the first stage after fathoming. The cost calculations are presented in Table 7. As a result of fathoming, only 10 states out of the

Table 6
Stage 1 DP fathoming calculations

State	S	i	Slot assignment	Updated $f_k(S,i)$	$q(S,i)$	$C(S,i)$	LB	$u(S,i)$	Improved $u(S,i)$	UB
1	A1	A2	__ __ A2 A1	400	555	345	1300	1375	1350	1350
2	A1	A3	__ __ A3 A1	400	520	415	1335	1380	1350	1350
3	A1	A4	__ __ A4 A1	415	600	335	1350	–	–	1350
4	A2	A1	__ __ A1 A2	360	555	390	1305	1380	1350	1350
5	A2	A3	__ __ A3 A2	360	565	415	1340	1380	1350	1350
6	A2	A4	__ __ A4 A2	370	635	305	1310	1370	1370	1350
7	A3	A1	__ __ A1 A3	425	520	390	1335	1380	1355	1350
8	A3	A2	__ __ A2 A3	425	565	345	1335	1375	1355	1350
9	A3	A4	__ __ A4 A3	395	600	335	1330	1380	1375	1350
10	A4	A1	__ __ A1 A4	350	600	405	1355	–	–	1350
11	A4	A2	__ __ A2 A4	320	635	355	1310	1370	1370	1350
12	A4	A3	__ __ A3 A4	350	600	385	1335	1385	1370	1350

Table 7
Stage 2 DP cost calculations

State	S	i	j	$\theta_{ij}(S)$	$f_{k-1}(S\backslash\{j\},j)$	Updated $f_{k-1}(S\backslash\{j\},j)$	Total cost	$f_k(S,i)$
1	A1, A2	A3	A1	390	395	360	750	745
			A2	345	450	400	745	
2	A1, A2	A4	A1	420	395	365	785	780
			A2	350	450	430	780	
3	A1, A3	A2	A3	415	435	400	815	815
			A1	390	445	425	815	
4	A1, A3	A4	A3	385	435	415	800	800
			A1	405	445	395	800	
5	A2, A4	A1	A2	350	345	330	680	670
			A4	305	405	365	670	
6	A2, A4	A3	A2	355	345	320	675	675
			A4	305	405	370	675	
7	A2, A3	A1	A2	345	485	425	770	770
			A3	415	400	360	775	
8	A2, A3	A4	A2	355	485	435	790	790
			A3	425	400	370	795	
9	A3, A4	A1	A3	385	365	350	735	730
			A4	335	455	395	730	
10	A3, A4	A2	A3	425	365	320	745	740
			A4	305	455	435	740	

potential 12 are present. For states 3, 4 and 6, the two possible slot assignments result in the same value of $f_k(S,i)$, implying that there may be multiple optimal solutions.

As part of the process at stage k, the cost function $f_{k-1}(S\backslash\{j\},j)$ that was calculated in stage $k-1$ is updated now that the slot position of new flight i is known. Before the position of flight i was known, it was assumed that passengers arriving on flights in the set S would miss their connecting flights (outbound flights whose predecessor is flight i). The calculated value of $f_{k-1}(S\backslash\{j\},j)$ was based on this assumption. A comparison of the data in columns 6 and 7 of Table 7 show that the missed connection costs were overestimated by as much as 10%.

Table 8
Stage 2 DP fathoming calculations

State	S	i	Slot assignment	Updated $f_k(S,i)$	$q(S,i)$	$C(S,i)$	LB	$u(S,i)$	UB
1	A1, A2	A3	__ A3 A2 A1	745	210	395	1350	–	1350
2	A1, A2	A4	__ A4 A2 A1	780	290	305	1375	–	1350
3	A1, A3	A2	__ A2 A3 A1	815	210	340	1365	–	1350
4	A1, A3	A4	__ A4 A3 A1	800	255	325	1380	–	1350
5	A2, A4	A1	__ A1 A4 A2	670	290	405	1365	–	1350
6	A2, A4	A3	__ A3 A2 A4	675	290	405	1370	–	1350
7	A2, A3	A1	__ A1 A2 A3	770	210	375	1355	–	1350
8	A2, A3	A4	__ A4 A2 A3	790	290	295	1375	–	1350
9	A3, A4	A1	__ A1 A4 A3	730	255	390	1375	–	1350
10	A3, A4	A2	__ A2 A4 A3	740	290	350	1380	–	1350

Table 9
Results from pairwise exchange procedure

Slot assignment	Total cost, $
A4 A3 A2 A1	1350
A3 A4 A2 A1	1375
A4 A2 A3 A1	1365
A4 A3 A1 A2	1355

Table 8 gives the fathoming results for the second stage. Because LB $\geqslant$ UB, all states are fathomed and the algorithm terminates with UB = \$1350. Looking back at stage 1 where UB was first set to \$1350, we see that the corresponding slot assignments are (A4, A3, A2, A1).

The final step is to perform pairwise exchanges to see if the current solution can be improved. Table 9 lists possible assignments that can be achieved by interchanging any two neighboring slots at a time. The results indicate that none of the possibilities improves upon the incumbent cost of \$1350 that was found by the algorithm so this is the optimum.

8. Computational results

The solution procedure described in Section 6 was coded in JAVA. All computations were performed on a 3.2 GHz Intel Pentium 4 processor running on Microsoft Windows XP Professional Version 2002. The assignment problems were solved with the shortest augmenting path algorithm of Jonker and Volgenant (1987).

8.1. Experiment with real data

The first set of experiments was conducted using the GDP issued to American Airlines on November 8, 1993 at DFW airport by the FAA (see Luo and Yu, 1997). Minor changes were made to the data set to adapt it to the framework of our DP algorithm. In all, the GDP involved 71 flights and spanned three hours in duration.

After running the algorithm for more than two hours without any signs of convergence, we decided that a better understanding of its limits was required. Accordingly, we created several smaller instances starting with six flights, and reran the algorithm to determine the largest problem that could be solved in a reasonable amount of time. The original flight schedules, and the delay and missed connection costs formed the input.

The metrics used in the computations are defined in Table 10. The initial solution cost is the cost associated with the initial assignments given by the FAA to the airline. The final solution cost is the cost produced by the algorithm upon termination.

For the preliminary testing, three data sets for problem instances ranging in size from 6 to 15 flights were generated and solved. Table 11 presents the average results. In addition to running the algorithm, complete enumeration was performed on all but the 15-flight instances to determine the optimal costs and the total number of feasible states. The third column in Table 11, 'Initial cost savings (%)', refers to the cost savings

Table 10
Metrics used in computational experiments

Metric	Definition
Cost savings, %	((initial solution cost − final solution cost)/initial solution cost)) × 100
Optimality gap, %	((final solution cost − optimal solution cost)/optimal solution cost)) × 100
States searched, %	(number of states evaluated/number of feasible states) × 100
States fath/eval, %	(number of states fathomed by lower bound/number of states evaluated) × 100
States fath/feas, %	(number of states fathomed by lower bound/number of feasible states evaluated) × 100

Table 11
Summary of results of preliminary tests

Problem size	No. feasible states	Initial cost savings (%)	Final cost savings (%)	Optimality gap (%)	States searched (%)	States fath/eval (%)	States fath/feas (%)	Average time (s)
6 flights	192	7.9	12.4	0.5	29.2	83.1	24.3	0.1
8 flights	1024	14.4	24.6	1.5	15.2	83.1	12.6	0.1
10 flights	4561	4.1	7.3	0.4	5.7	86.5	4.9	0.2
12 flights	24,576	5.0	10.0	1.0	7.5	90.5	6.7	1.5
15 flights	–	4.8	16.6	–	–	77.1	–	46

after only pairwise slot exchanges were performed on the initial solution provided by the FAA. Approximately 50% of the final total cost savings were achieved by the pairwise slot exchanges.

On average, 65 min were required to solve the 12-flight instances by enumeration but only 1.5 s with the algorithm, which produced a solution that was within 1% of the optimum on average. The largest gap was 2.7%. All instances were solved within a minute with the algorithm, the longest being one of the 15-flight instances which took 58 s. The average solution times are reported in the last column of Table 11. When the problem size was increased to 20 flights, the algorithm failed to converge within an hour. From a practical point of view, about 20 min is all that an airline can allow for the computations.

Considering this constraint, it was necessary to reduce the run time at the expense of solution quality. In the implementation, a rolling horizon framework was used in which problems with more than 15 flights are broken down into subproblems of size 15 (or less). When solved, the latter 10 flights were fixed and the process repeated with 10 new flights and the remaining 5 that were not fixed. With this approach, the accuracy demonstrated by the results in Table 11 can only be expected for each subproblem and not for the original problem as a whole.

Fig. 3 illustrates how a problem with 70 flights would be solved using a 15/10 rolling horizon. Slots 56–70 and the flights assigned to them in the initial GDP form the first subproblem. When a solution is obtained the flights assigned to slots 61–70 by the algorithm are fixed. Flights assigned to slots 56–60, as well as the flights assigned to slots 46–55 in the initial GDP, form the second subproblem associated with slots 46–60. The process continues until all flights are assigned. At that point, pairwise exchanges are performed on the entire schedule to improve the solution. Thus, a 70-flight problem is solved as seven subproblems, each with a maximum of 15 flights.

In an attempt to further analyze the performance of the algorithm on different size problems, the original data set was broken down into instances ranging from 20 to 70 flights. The results obtained with the rolling horizon approach are reported in Table 12. Somewhat surprisingly, even though the algorithm had to solve far fewer subproblems for the 20-flight instance, the run time for this case was up approximately 100% higher than for all others. A closer look at the calculations provided some insights, but before offering an explanation, let us look at the reasons why one problem may be more difficult to solve than another.

First, more feasible slot allocations lead to more feasible states for the algorithm to evaluate. The number of ways that slots can be assigned depends on the scheduled arrival times, the GDP arrival slots, and the number of flights from the same origins. Second, the lower bound calculation does not include the cost of missed

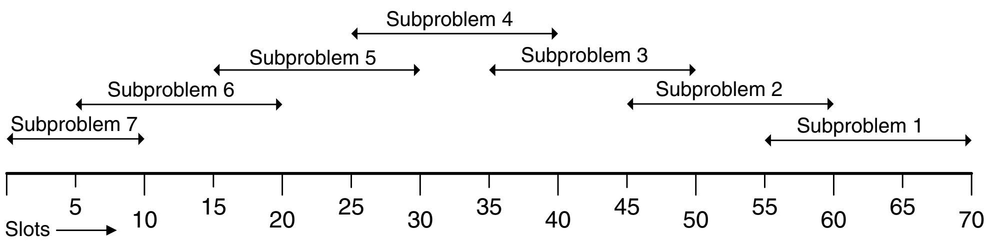

Fig. 3. Example of rolling horizon framework.

Table 12
Results for real data set

Problem size	States generated	Cost savings (%)	States fathomed (%)	Total time (s)
20	11,183	6.0	74.7	220
30	8561	11.8	76.7	108
40	8732	11.3	77.3	108
50	9841	14.8	78.3	112
60	9731	13.3	78.8	109
70	10,953	13.6	79.6	112

connections to outbound flights whose predecessors are in the GDP, as mentioned in Section 6. If the magnitude of that cost is relatively large, then the lower bounds provided by solving the assignment problem will not be good enough to fathom many states and the value of branch and bound will be marginal. Comparatively speaking, the number of missed connections decreases as the number of flights in the GDP that do not have outbound flights increases.

Third, the spacing of arrival slots also has an impact on the number of missed connections and thus problem difficulty. For example, let us consider a 10-flight instance where the first and last time slots are 10:00 and 10:10, respectively. When the aircraft turnaround time is 30 min, a flight assigned to slot 10:00 will be ready to leave for its outbound flight by 10:30. The earliest time any of the outbound flights can leave is 10:30. Assuming a 20-min turnaround time for passengers, someone arriving on a flight that is assigned to the last slot will be ready to board a flight by 10:30. More generally, the latest time that passengers arriving on any of the GDP flights would be ready to board an outbound flight is 10:30, which implies that there will not be any missed connections between flights in the GDP at all. Thus, the spacing of slots plays a vital role in determining the number of missed connections and corresponding costs that, in turn, affect the rate that states are fathomed.

Returning to the results in Table 12, the subproblems associated with the 20-flight instance were the most difficult to solve and hence took significantly more time. Although those 20 flights are included in the other five instances, the subproblems in each instance were different. The 30-flight instance, for example, had three subproblems whereas the 20-flight instance had only two. For the first subproblem that had to be solved when 30 flights were in the GDP, the difference between the first and the last arrival slots was only 15 min. Moreover, six of the 15 flights did not have outbound flights. The second subproblem was identical to the first subproblem for the 20-flight instance with respect to arrival slot times, but the actual flights being considered were slightly different. The subproblem for the 20-flight instance had more feasible slot allocations. In fact, that particular subproblem was the most difficult in the entire data set. As a consequence, the smallest problem took more time to solve than any of the larger ones.

As a final point, suppose that we start with a 25-flight problem and add 5 more flights to get a 30-flight problem. In either case, the corresponding subproblems may be different enough that the larger problem turns out to be easier to solve. To summarize, the difficulty of a problem depends on the subproblems created when using a rolling horizon, and because the nature of the subproblems can be very different, there may not be a strong correlation between number of flights and computational effort required by the DP algorithm.

8.2. Summary of experiments

The American Airlines data set is typical in terms of its size. Although larger instances may arise in practice (Luo and Yu, 1997), they can often be decomposed into independent groups because flights are arranged in complexes and large gaps may exist between groups of flights. For testing purposes, we generated four additional data sets with durations 02:00, 02:15, 02:30 and 02:45 h using the American data set as a guide. Flight schedules were created manually and costs were assigned to reflect relative values reported in the literature.

As mentioned, pairwise slot exchanges were performed on the full schedule after all the subproblems were solved to improve the solution. The cost savings observed for each of the data sets with respect to the initial GDP and the averages for each problem size are given in Table 13a before the pairwise exchanges were performed. Table 13b presents the total cost savings after the pairwise slot exchanges were performed.

Table 13a
Cost savings before pairwise exchanges for all data sets (%)

Problem size	Dataset 1	Dataset 2	Dataset 3	Dataset 4	Dataset 5	Average
20	6.0	4.9	12.8	6.2	4.1	6.8
30	11.8	6.2	9.8	1.3	3.7	6.6
40	10.5	8.9	9.4	5.6	4.9	7.9
50	14.8	10.4	8.1	8.5	7.3	9.8
60	13.1	10.3	11.4	7.6	8.7	10.2
70	13.4	8.9	9.5	8.5	9.0	9.9

Table 13b
Cost savings after pairwise exchanges for all data sets (%)

Problem size	Data set 1	Data set 2	Data set 3	Data set 4	Data set 5	Average
20	6.0	5.6	12.8	6.9	4.1	7.1
30	11.8	6.7	11.0	2.9	3.8	7.2
40	11.3	9.2	10.7	6.6	5.6	8.7
50	14.8	10.9	9.3	10.0	7.8	10.6
60	13.3	10.7	12.1	8.9	9.4	10.9
70	13.6	10.0	10.2	9.8	9.9	10.8

Comparing the last columns in Tables 13a and 13b, we see that on average, there was a 0.7% cost savings due to the pairwise exchanges. The last column in Table 13b is plotted as a function of problem size in Fig. 4. The curve illustrates that as more flights are included in the GDP, the cost savings first increases but then levels out at around 10% or 11%.

The rates at which states are fathomed by the lower bound are presented in Table 14. The average rate for all instances was about 83% and did not vary significantly across problem sizes.

Fig. 5 plots the average, minimum, and maximum times that it took for the algorithm to solve the various problem instances. As discussed previously, the computational effort depends strongly on the subproblems generated, and for these instances at least, there is no direct relationship with problem size.

To determine the effect of missed connection costs on problem difficulty, a parametric analysis was performed by varying the costs associated with the outbound flights whose predecessors are not in the GDP. Fig. 6 plots the cost savings (%) when those missed connection costs were magnified by a factor of 2 and then 5. No consistent trends were observed with respect to the solution time or the rate at which states were fathomed. When the missed connection costs associated with the outbound flights whose predecessors are in the GDP were varied, the results were similar. Thus, no relationship could be found between this aspect of cost and the performance of the algorithm. Results of the parametric analysis are provided by Mohan (2005).

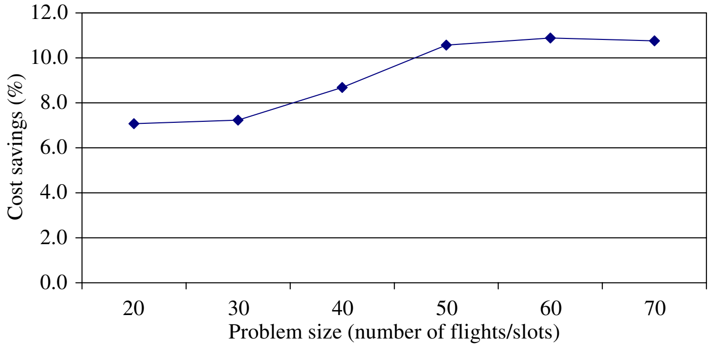

Fig. 4. Average cost savings.

Table 14
States fathomed (%)

Problem size	Average no. of states	Dataset 1	Dataset 2	Dataset 3	Dataset 4	Dataset 5	Average
20	3548	74.7	82.2	85.8	92.0	90.6	85.1
30	4792	76.7	89.7	78.8	86.1	88.5	84.0
40	5174	77.3	87.1	86.4	94.9	85.1	86.1
50	8708	78.3	82.7	81.1	87.3	82.3	82.3
60	11,952	78.8	73.9	82.3	84.8	81.1	80.2
70	11,016	79.6	83.7	83.6	86.2	78.7	82.4

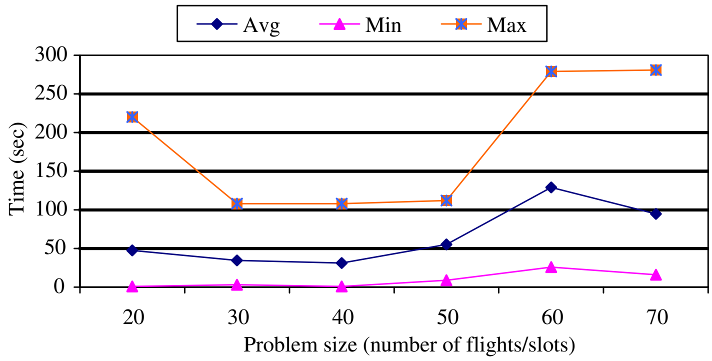

Fig. 5. Average solution times.

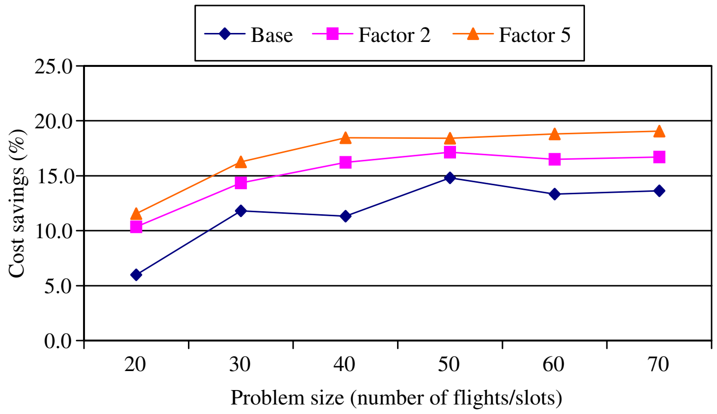

Fig. 6. Parametric analysis.

9. Summary and conclusions

In this paper, the arrival slot reallocation problem was modeled as a DP with the objective of assigning the flights in the GDP to minimize total cost to an airline. Although the delay and fuel costs were assumed to be linear, the fact that the missed connection costs depended on the slots assigned to all flights in the GDP, precluded the use of a standard linear model for the problem. To find solutions, a branch and bound scheme was introduced in the DP framework to reduce the number of states. Testing showed that problems with up to 12 flights could be solved within a minute to within 1% of optimality on average, and problems with as many as 15 flights could be solved to about 2.7% of optimality in the same amount of time. Because larger instances ran well beyond the industry-imposed time limit (as we understand it) of 20 min, it was necessary to sacrifice some amount of solution quality to realize greater computational efficiency. This was achieved by embedding the DP in a 15/10 rolling horizon structure in which large instances were split into subproblems of size 15 flights or less and the latter 10 flights were fixed in accordance with the solution. After all the subproblems were solved, pairwise slot exchanges were performed on the incumbent solution in an effort to reduce the overall cost.

In addition to the real data associated with an FAA issued GDP for American Airlines, four problem instances were randomly generated and solved to further analyze the effectiveness of the proposed methodology. In all cases, solutions were found within 5 min that were 9% on average below the cost associated with the initial GDP. This phase of the testing revealed that the number of flights alone is not a strong determinant of problem difficulty. Other factors, such as the spacing of the arrival slots in the GDP and the effectiveness of the lower bounding scheme, are really what determine the size of the states space and hence are more critical in this regard.

To reduce the size of the state space, a stronger lower bounding procedure is needed. If a more effective way can be found to model the missed connection costs to outbound flights whose predecessors are in GDP, then more states could be fathomed. This means that much larger problem instances could be solved either with or without the rolling horizon framework, bringing us closer to achieving the accuracy demonstrated in the preliminary testing stages of the algorithm. One way to provide an improvement in the calculation of the assignment problem (5) cost coefficients c_{jm} is to assign a flight r in the GDP to slot $m-1$ and then determine the corresponding missed connection costs for passengers arriving on flight j and connecting to outbound flight $\phi(r)$. To ensure that the corresponding lower bound is still valid, it would be necessary to consider all flights $r \in N_i \backslash S$, $r \neq j$ and select the one that gave the minimum cost. We tried this idea in the development stage of our lower bounding procedure but realized little improvement. Moreover, the computational burden increased by about 15% due to the fact that c_{jm} has to be calculated for each state (S, i).

Finally, we note again that a potentially promising alternative to (5) is to model the lower bounding problem as a QP. Very small instances should be solvable quickly; for larger instances, a relaxed solution might still be useful for fathoming states. Of course, solving a relaxation of the full $n \times n$ QP would provide a lower bound on the true optimum to the GDP. These ideas are left to future research.

References

Andreatta, G., Romanin-Jacur, G., 1987. Aircraft flow management under congestion. Transportation Science 21 (4), 249–253.

Bard, J.F., Yu, G., Argüello, M.F., 2001. Optimizing aircraft routings in response to groundings and delays. IIE Transactions on Operations Engineering 33 (10), 931–947.

Chang, K., Howard, K., Oiesen, R., Shisler, L., Tanino, M., Wambsganss, M.C., 2001. Enhancements to the FAA ground-delay program under collaborative decision making. Interfaces 31 (1), 57–76.

Erdoğan, G., Tansel, B., 2007. A branch-and-cut algorithm for quadratic assignment problems based on linearizations. Computers and Operations Research 34 (4), 1085–1106.

Jonker, R., Volgenant, A., 1987. A shortest augmenting path algorithm for dense and sparse linear assignment problems. Computing 38 (4), 325–340.

Luo, S., Yu, G., 1997. On the airline schedule perturbation problem caused by the ground delay program. Transportation Science 31 (4), 298–311.

Luo, S., Yu, G., 1998. Airline schedule perturbation problem: landing and takeoff with nonsplittable resource for the ground delay program. In: Yu, G. (Ed.), Operations Research in the Airline Industry. Kluwer, Boston, pp. 404–432 (Chapter 14).

Mehndiratta, S.R., Keifer, M., 2003. Impact of slot controls with a market-based allocation mechanism at San Francisco International Airport. Transportation Research Part A 37 (7), 555–578.

Mohan, D.N., 2005. Solving the arrival slot reallocation problem during a ground delay program. MS Thesis, Graduate Program in Operations Research and Industrial Engineering, The University of Texas, Austin.

Pisinger, W.D., Rasmussen, A.B., Sandvik, R., 2007. Solution of large quadratic knapsack problems through aggressive reduction. INFORMS Journal on Computing 19 (2), 280–290.

Rakshit, A., Krishnamurthy, N., Yu, G., 1996. System operations advisor: A real-time decision support system for managing airline operations at United Airlines. Interfaces 26 (2), 50–58.

Richetta, O., Odoni, A.R., 1993. Solving optimally the static ground-holding policy problem in air traffic control. Transportation Science 27 (3), 228–238.

Richetta, O., Odoni, A.R., 1994. Dynamic solution to the ground-holding problem in air traffic control. Transportation Research Part A 28 (3), 167–185.

Rossi, F., Smriglio, S., 2001. Set packing model for the ground holding problem in congested networks. European Journal of Operational Research 131 (2), 400–416.

Ryan, A.J., 2001. Human factors and collaborative decision making. http://mason.gmu.edu/~ajryan/research.html.

Stojkovic, G., Soumis, F., Desrosiers, J., Solomon, M.M., 2002. An optimization model for a real-time flight scheduling problem. Transportation Research Part A 36 (9), 779–788.

Teodorovic, D., 1985. A model for designing the meteorologically most reliable airline schedule. European Journal of Operational Research 21 (2), 156–164.

Teodorovic, D., Stojkovic, G., 1995. A model to reduce airline schedule disturbances. Journal of Transportation Engineering 121 (4), 324–331.

Terrab, M., Odoni, A.R., 1993. Strategic flow management for air traffic control. Operations Research 41 (1), 138–152.

Thengvall, B.G., Yu, G., Bard, J.F., 2001. Multiple fleet aircraft schedule recovery following hub closures. Transportation Research Part A 35 (4), 289–308.

Vasquez-Marquez, A., 1991. American airlines arrival slot allocation system (ASAS). Interfaces 21 (1), 42–61.

Vranas, P.B., Bertsimas, D.J., Odoni, A.R., 1994a. The multi-airport ground-holding problem in air traffic control. Operations Research 42 (2), 249–261.

Vranas, P.B., Bertsimas, D.J., Odoni, A.R., 1994b. Dynamic ground-holding policies for a network of airports. Transportation Science 28 (4), 275–291.

Yan, S., Lin, C., 1997. Airline scheduling for the temporary closure of airports. Transportation Science 31 (1), 72–82.

Yan, S., Tu, Y., 1997. Multi-fleet routing and multi-stop flight scheduling for schedule perturbation. European Journal of Operational Research 103 (1), 155–169.

Yan, S., Yang, D., 1996. A decision support framework for handling schedule perturbation. Transportation Research Part B, Methodological 30 (6), 405–419.

Yan, S., Young, H., 1996. A decision support framework for multi-fleet routing and multi-stop flight scheduling. Transportation Research Part A, Policy and Practice 30 (5), 379–398.