Linköping University Post Print

The flight perturbation problem

Tobias Andersson Granberg and Peter Värbrand

N.B.: When citing this work, cite the original article.

This is an electronic version of an article published in:

Tobias Andersson Granberg and Peter Värbrand, The flight perturbation problem, 2004, Transportation planning and technology, (27), 2, 91-117.

Transportation planning and technology is available online at informaworldTM:

<http://dx.doi.org/10.1080/0308106042000218195>

Copyright: Taylor & Francis

<http://www.tandf.co.uk/journals/default.asp>

Postprint available at: Linköping University Electronic Press

<http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-17061>

The Flight Perturbation Problem

T. Andersson*

University of Linköping, ITN
SE-601 74 Norrköping, Sweden
email: toban@itn.liu.se

P. Värbrand

University of Linköping, ITN
SE-601 74 Norrköping, Sweden

* corresponding author

Abstract

Airlines spend considerable time, effort and financial resources on planning. It is essential to create a competitive timetable and construct a fleet and a crew schedule that utilizes these resources to the maximum. Unfortunately, it is all too common that an airline is faced with the necessity of reconstructing their schedules due to some unforeseen event, e.g., an aircraft breakdown or a crew member that is indisposed. In this paper, an application that can help airlines solve the complex problem of reconstructing aircraft schedules is presented. A mixed integer multicommodity flow model with side constraints is developed and further reformulated into a set packing model using the Dantzig-Wolfe decomposition. Cancellations, delays and aircraft swaps are used to resolve the perturbation, and the model ensures that the schedule returns to normal within a certain time. Two column generation schemes for heuristically solving the model are tested on real problem data obtained from a Swedish domestic airline. The computational tests show that the application is capable of presenting high quality solutions in a few seconds and therefore can be used as a dynamic decision support tool by the airlines.

Keywords: Airline operations; Scheduling; Optimization; Perturbations; Dantzig-Wolfe decomposition

1 INTRODUCTION

Airlines handle perturbations in their flight schedules on a daily basis. The personnel responsible for operational planning can normally be found in airline operational control (AOC), and their main task is to ensure the safety and efficiency of each flight in the timetable. The ideal solution is of course to keep to the original plan, but unfortunately this is not always possible. Sometimes there is a perturbation in the plan, due to for example:

- **Aircraft malfunction:** Anything that compromises the safety of a flight will have to be dealt with immediately. A malfunction may be more or less serious, but the result is that the aircraft may not be able to operate any flights for some time.
- **Absent crew:** As a complement of crew is needed to operate a flight, an absent crew member may cause serious problems. The absence may be due to illness, traffic congestion, delays on previous flights, or any other non-planned reason.
- **Weather conditions:** Ice on the runway, fog or a powerful headwind are examples of how weather may cause problems for an airline. The perturbation can be minor and cause a delayed departure, or so severe that an airport has to be closed down.
- **ATC related:** The air traffic control authorities have the responsibility for controlling their airspace, ensuring that all airlines can operate their flights safely. They have the power to intervene in the plans made by airlines and, for example, delay departure times or force aircraft into holding patterns leading to delayed arrival times.

The list can be made much longer, but for the personnel at the AOC the reason for the perturbation is not as important as taking care of the effects emanating from it. The personnel at the AOC are divided into different groups. The flight controllers handle the aircraft and are usually the ones who decide first on which action to take when a perturbation occurs. Before doing anything they have to check with the crew controllers and the maintenance planners that the action considered will not cause any serious problems in their areas of responsibility. There are four basic options available for the flight controllers:

- **Delaying:** Delaying the departure of a flight will directly affect the passengers on that particular flight. It may also, indirectly, affect the passengers on the next flight in the route for the aircraft in question, if the planned ground time between the two flights is too short to cover for the delay.
- **Swapping:** If an aircraft is assigned to a flight not in its original route, a swap has been performed. If the flight swap is made to an aircraft of another type, it is important to check how the capacity of the aircraft matches the number of passengers on the flight. A swap may also lead to problems in crew planning, since many crew members are only certified on one aircraft type.
- **Cancelling:** To cancel a flight is maybe the most extreme option from the passenger's point of view, unless there are alternative ways of transporting passengers to their destination (a cheaper solution for an airline is often to let them take the next available flight). Furthermore, cancelling a flight means a serious disruption in the route of the aircraft assigned to it. This aircraft now has a difficult time in getting to the correct airport for the next flight in the route, often making it necessary to cancel more than one flight.
- **Positioning:** To fly an aircraft without any passengers between two airports is a positioning or ferrying flight. This happens when the aircraft has to be at a certain airport at a certain time, and there is no regular flight the aircraft can operate to get there. This is an expensive solution for the airline, partly because of the flying cost

(fuel, personnel, etc.), but also because the aircraft is unavailable during the positioning time, which is why positioning flights almost always are done during the night.

The ‘flight perturbation problem’ (FPP) can be briefly stated as: *Minimize the negative consequences of a perturbation that has made it impossible for one or more flights to depart on their scheduled time operated by their originally planned aircraft.* The negative consequences are of course the delays, swaps, cancellations and positioning flights needed to solve the problem. However, the consequences are also the impact the solution has on maintenance and crew planning. The crew planning will most likely have to be altered to fit the new schedule, and if this cannot be done in a satisfactory way, the crew planners will reject the solution. Likewise, if the solution is infeasible from a maintenance perspective, it is unacceptable.

The ideal approach would be to solve the FPP while guaranteeing that the maintenance and crew planning stays feasible. There are, however, other maybe more important qualities that an application has to possess to be of use for a flight controller. One of the most important qualities is that it has to produce a solution fast. The disruptions in the schedule will become more serious with time if no changes are made to prevent this from happening. Thus, the controllers have to make very quick decisions, in most situations, and cannot wait a long time for a solution. It is difficult to state a minimum response time, but as a ‘rule of thumb’, it should be within a few seconds.

Another important quality is that the solutions should be easy to understand and intuitively plausible from a user perspective, otherwise the controller may dismiss the solution simply because it does not appear valid. A solution should therefore have as few changes as possible from the original schedule. It should also guarantee that the schedule returns to normal after a certain time.

Teodorovic et al. [13] were one of the first to study the problem from an operations research perspective. They considered a situation where an aircraft is taken out of service and try to minimize total passenger delay by swapping and delaying flights. The proposed model is solved exactly by a branch and bound (B&B) method. The concept is further developed in Teodorovic et al. [14] where the same problem, i.e., an aircraft shortage, is considered. A lexicographic model allowing cancellations, delays and swaps is solved heuristically with an algorithm based on dynamic programming. The main objective is to minimize cancellations, but if there exist several solutions with the same number of canceled flights, the second objective function minimizes total passenger delay. In a third paper Teodorovic et al. [15] propose a more complete solution process to solve the FPP, including the consideration of both maintenance and crew. The problem is solved sequentially and heuristically by first making a new crew schedule after which a new aircraft schedule is created and checked for maintenance feasibility. The same lexicographic objectives as in [14] are used, although no mathematical model is presented. Still, less serious perturbations are considered but beside aircraft shortages that were considered in the earlier papers, canceled crew and changes in planned departure time are treated as well.

Jarrah et al. [8] studied the aircraft shortage problem and presented two network flow models; one that solves the problem by delaying and swapping flights (at a single airport), and the other by cancelling and swapping flights. In both models it is also possible to use spare aircraft, but neither maintenance nor crew considerations were implemented. The objective in the models is to minimize the costs associated with the recovery, i.e., the swap and ferry costs in both models, as well as the delay costs in the first model and cancellation costs in the second. Both models are minimum-cost flow network problems and can be efficiently solved to optimality. The models of Jarrah et al. [8] were utilized in the System Operations Advisor,

used by United Airlines. Rakshit et al. [12] described the interface and how the advisor is used. Another paper focusing on the interface of a decision support system for airline operations control is proposed in Mathaisel [10].

Cao et al. [3,4] extended the delay model of Jarrah et al. [8] to incorporate cancellations and multiple airports. A quadratic 0-1 programming model is presented, in which the flight revenue with subtracted swap and delay costs are maximized. It is not clear which kind of perturbations were studied, since the computational experiments start with a randomly generated non-optimal schedule. This schedule is then improved heuristically, without regard for maintenance and crew planning.

Yan et al. [18,19] developed four models of increasing complexity for dealing with schedule perturbation problems. In the first model, it is possible to cancel flights in order to repair the schedule. The second model allows both the cancellation of flights and the ferrying of spare aircraft. The third model considers cancellations and delays of flights and the last model incorporates all of the possible decisions previously considered, i.e., delays, cancellations and ferrying of spare aircraft. Flight swaps within the same fleet of aircraft are allowed in all models. The objective in all models is to minimize the cost for repairing the schedule, but passenger revenues are still included. The first two models are pure network flow models and quite easy to solve exactly. The other two models are network flow models with side constraints and solved heuristically by Lagrangian relaxation and subgradient optimization. The perturbation considered is the shortage of a single aircraft. Neither crew nor maintenance constraints are considered. In Yan et al. [19] a multifleet version of the model described above is presented, where a larger aircraft type can be assigned to a leg that originally was scheduled with a smaller aircraft type. Yan et al. [20] use the same model for solving the perturbation problem for the temporary closure of an airport. The difference is that, instead of an aircraft missing in the network, an airport is unavailable. The closure of an airport is an example of a very serious disruption, which often have to be solved by changing the arrival airports for the affected flights, i.e., letting the aircraft land at another airport and transfer the passengers to their destination. However, the model that is presented only allows for flight swaps, delays and cancellations, which mean that all flights arriving or departing from the closed airport have to be canceled or delayed.

Arguello et al. [1] have presented a greedy randomized adaptive search procedure to reschedule aircraft routings when there is an aircraft shortage. The heuristic allows cancellations, delays and flight swaps, while minimizing the total cost for delays and cancellations. Maintenance and crew considerations are not implemented.

Thengvall et al. [16] used the model proposed by Yan et al. [18,19] and extended it to incorporate the ability to penalize deviations from the original schedule. The model is capable of handling cancellations, delays and flight swaps, but does not consider crew or maintenance issues. The objective is to maximize passenger revenues while subtracting the delay costs. The perturbation considered is aircraft shortages, although the authors claim that other disruptions can easily be modeled as well. Integer solutions to the model are obtained heuristically, when the LP relaxation yields fractional solutions. Thengvall et al. continued their work in [17] where the more serious disruption of airport closures are considered. Like in Yan et al. [20], no destination changes are allowed, which means that the models consider the usual set of actions - cancellations, delays, swapping and ferrying.

Love et al. [9] developed a local search heuristic to solve the problem. Cancellations, delays and swaps between aircraft of the same type are allowed. The actions are weighted in the objective function, making it possible - by changing the weights - to obtain solutions with different characteristics in terms of the number of swaps, cancellations etc.

In this paper a maximum revenue multicommodity flow model with side constraints is developed for the flight perturbation problem, allowing cancellations, delays and flight swaps between different aircraft types. The model is reformulated into a set packing model, using the Dantzig-Wolfe decomposition, where each column represents a feasible route for a certain aircraft. Two column generation schemes, one based on LP relaxation and the other on Lagrangian relaxation, are used to find heuristic solutions quickly. The perturbations considered are of the less serious type, such as aircraft breakdowns and flight delays. This paper is a first attempt to use a column based model, i.e., the set packing model, to solve the FPP. There are many benefits with working with columns that represent aircraft routes, the foremost being that it makes it very easy to extend the model to incorporate new constraints. The focus in this paper is on the practical usability of the application that is developed. There has been little previous discussion in the literature on what kind of solutions are considered to be desirable from the intended users' point of view. This paper also considers how to find relevant objective coefficients, and the advantages and disadvantages of different kinds of coefficients.

In the next section, a network and two models are presented. In Section 3 the solution strategies are described followed by computational results in Section 4. Finally, conclusions and possible extensions to this research are discussed in Section 5.

2 A MATHEMATICAL REPRESENTATION OF THE FFP

When modelling the flight perturbation problem, a network that describes the current situation is useful. One possible network to use is the connection network shown in Figure 1. This network contains three different kinds of node: aircraft source nodes, flight nodes and flight sink nodes. Every node belongs to a station (i.e., an airport). Each aircraft source node represents a specific aircraft and belongs to the station where the corresponding aircraft is positioned at the start time, or where it will arrive if it is in the air at the start time. Each flight node, and flight sink node, represents a specific flight, and its position in the network represents the planned departure time and the departure station. To each flight sink node there is associated with it one unique aircraft source node. This means that any flow starting in a specific aircraft source node is forced to end in an associated flight sink node. The flight sink node represents a flight to which the aircraft, represented by the aircraft source node, was assigned in the original schedule. This forces the schedule back to its original form after the end time.

Between the start time and the end time is the decision time, during which it is possible to make changes in the schedule. The end time often coincides with the end of the day, since there is usually sufficient ground time during the night to cover any perturbations. A decision time of a day or less is usually sufficient for domestic flight schedules while a longer time might be necessary when dealing with long haul flights.

The arcs in the network represent feasible connections between sources, flights and sinks. If there is an arc between an aircraft source node and a flight node (or sink), the aircraft represented by the source node can be assigned to that flight. In the same way, an arc from flight node i to flight node (or sink node) j means that the aircraft that operated flight i can be assigned to flight j .

Figure 1. The connection network

The network can describe a situation without any disturbances as well as one in which something not planned for has occurred. If, for instance, there has been an aircraft breakdown, this will be represented by an aircraft source node appearing when the aircraft is repaired. If there is a delayed flight arrival, the source node will appear when the aircraft is available again, and a delayed flight departure (not aircraft related) will cause the associated flight node to appear later in time. A feasible solution in the network consists of one path from each aircraft source to the corresponding sink. The path will traverse a number of flight nodes; these are the flights the aircraft will operate. If a flight node is not traversed by any aircraft path, this flight will be canceled.

Thus far, three possible decisions when taking care of a perturbation are modeled by the network: aircraft swapping, flight cancellation and the use of spare aircraft. Delays can be modeled by moving the flight nodes in time. Ferrying of aircraft can be modeled by adding arcs from a given aircraft source node to the desired flight node. Among these decisions, the most difficult part to model is delay, since using delays as a dynamic decision tool means that continuous changes in the network structure are needed.

It is not unusual to treat all aircraft in one fleet as one commodity. However, this makes it harder to incorporate maintenance considerations, since it will not be possible to track a specific aircraft. If an aircraft is assigned to another sink node than originally planned, the flight assignments for this aircraft as well as all maintenance checks will be changed for the whole planning period, and not just during the decision period.

2.1 A Mixed Integer Multicommodity Flow Model

To capture and take advantage of the network structure, a mixed integer multicommodity flow formulation can be used as a basis for the mathematical model. Moreover, additional constraints taking the specific problem characteristics into account must be added. The Flight Perturbation Mixed Integer Multicommodity Flow Model (FP-MIMF) follows:

Sets:

A = set of aircraft source nodes.

F = set of nodes representing flights.

S = set of flight sink nodes.

Parameters:

c_{ij}^k = revenue gained if aircraft k is assigned to flight j after flight i .

c_i = cost per time unit for delaying flight i .

$s_i^k = 1$ if i is the correct source for k (i.e., $i = k$), else 0.

$t_i^k = 1$ if i is the correct flight sink for aircraft k , else 0.

AD_i = departure airport of flight i (represented by a number).

AA_j = arrival airport of flight j (represented by a number).

TD_j = planned departure time for flight j .

TA_i = planned arrival time for flight i (or when aircraft i is available).

TG_{ij}^k = required time between flights if aircraft k is assigned to flight j after flight i .

D_i = maximum allowed delay for flight i .

C^k = capacity of aircraft k .

P_j = number of passengers on flight j .

M = a large positive number.

Variables:

$x_{ij}^k = 1$ if aircraft k is assigned to flight j after having operated flight i , else 0.

d_i = how much the departure of flight i is delayed.

Maximize

$$\sum_{i \in A \cup F} \sum_{j \in F \cup S} \sum_{k \in A} c_{ij}^k x_{ij}^k - \sum_{i \in F} c_i d_i \quad (1)$$

Subject to

$$\sum_{j \in F \cup S} x_{ij}^k = s_i^k \quad i \in A, k \in A \quad (2)$$

$$\sum_{j \in F \cup S} x_{ij}^k - \sum_{j \in A \cup F} x_{ji}^k = 0 \quad i \in F, k \in A \quad (3)$$

$$\sum_{k \in A} \sum_{j \in F \cup S} x_{ij}^k \leq 1 \quad i \in F \quad (4)$$

$$\sum_{j \in A \cup F} x_{ji}^k = t_i^k \quad i \in S, k \in A \quad (5)$$

$$TA_i + TG_{ij}^k + d_i - (TD_j + d_j) + M(x_{ij}^k - 1) \leq 0 \quad i \in A \cup F, j \in F \cup S, k \in A \quad (6)$$

$$x_{ij}^k (AD_i - AA_j) = 0 \quad i \in A \cup F, j \in F \cup S, k \in A \quad (7)$$

$$d_i \leq D_i \quad i \in F \quad (8)$$

$$x_{ij}^k P_j \leq C^k \quad i \in A \cup F, j \in F, k \in A \quad (9)$$

$$x_{ij}^k \in \{0,1\} \quad \forall i, j, k \quad (10)$$

$$d_i = 0 \quad i \in A \cup S \quad (11)$$

$$d_i \geq 0 \quad i \in F \quad (12)$$

The constraints in (2) initiate a flow of one unit from each aircraft source, and in (3) are the flow conservation constraints. The constraints in (5) make sure that the aircraft originally assigned to a certain flight sink is the one that will be assigned to it in the solution. To have one flow variable for each commodity (aircraft) enables a strict cost control, e.g., in most cases small deviations from the original schedule are desired, which means that there will be a higher cost for operating a flight with another aircraft than originally planned. Constraint (4) makes sure that each flight node is traversed by at most one aircraft.

In order for an arc to be open, two conditions have to be met: flight j has to depart from the airport where flight i arrives and the departure time of flight j has to be later than the arrival time plus necessary ground time for flight i . The second condition may be violated though, by delaying the departure time for flight j . This will of course affect the arrival time of flight j , which may affect the departure time of flight l and so on. It is assumed that the required flight time between two airports is constant. The conditions discussed above are expressed in constraints (6) and (7), together with upper bounds on the delay variables in (8).

Constraint (9) makes sure that x_{ij}^k can take the value one only if the capacity of the aircraft is greater than the number of passengers on flight j . It should be noted that this constraint, as well as constraint (7), are bounds leading to a fixation if active, and may therefore be omitted in the model formulation.

There are principally two ways of setting costs: by using real costs or weights. When using real costs, a c_{ij}^k should include the net revenue for operating flight j , the cost for using a new aircraft if k was not supposed to fly j in the beginning (i.e., a swap cost) and maybe a transition cost from flight i to flight j . The cost for the delays, c_i , should include a ‘badwill’ cost as well as costs for reassigning and compensating passengers. Many of these costs are of course very hard to obtain, e.g., a badwill cost can never be more than an estimate. The main advantage with using real costs, if such can be found, is that it is easier for the flight controller to evaluate the quality of different solutions.

When weights are used, the controller will try to find a valid trade-off between the actions, e.g., weight the cost for aircraft swaps within the same fleet to 1, between different aircraft types to 20, for cancelling a flight to 10 and for delaying a flight to 1. One practical way to obtain the revenue c_{ij}^k from these weights is to start by letting the revenue for operating flight j equal the cancellation weight multiplied by the number of passengers on that flight. Thus, the revenue in the model is not a measure of how much the passengers pay for their tickets, but more of an inverted shortage cost for not operating a certain flight. From this revenue the swap weight is subtracted if aircraft k was not originally assigned to that flight. It should be noted that the swap weights are the components in the model that effectively preserve large parts of the original schedule. The cost c_i is calculated as the weight for delays multiplied by the number of passengers on flight i .

An obvious advantage with weights is that the controller can easily modify a weight and re-optimize the solution. Hence, the experience of the controller can be utilized efficiently, as he or she is able to judge the solutions produced by the model, and add relevant aspects that are not covered. Thus, the model can be used as an interactive decision support system, not telling the user what should be done, but giving suggestions on possible good solutions.

2.2 Dantzig-Wolfe Decomposition of the MIP Model

Without constraints (4) in FP-MIMF, the model decomposes into one problem for each aircraft. To separate the delays over the aircraft, let d_i^k be the delay on flight i caused by aircraft k . Since at most one aircraft can be assigned to a flight, we have:

$$d_i = \sum_{k \in A} d_i^k \quad i \in F \quad (13)$$

as, for all k , only one d_i^k can be nonzero. For one aircraft k , there may however exist two feasible solutions, where the values of x_{ij}^k are equal but the values of d_i^k differs. This happens when it is possible to delay a flight for a certain time without affecting the flights later in the route. The solution with the extra delay is clearly dominated by the other solution, since an extra delay will lower the objective function value. Therefore, it is never interesting to consider solutions where the delay is more than required by constraint (6).

Let R^k be the set of feasible solutions for aircraft k that is not dominated by any other solution and let r be a point in the set. Furthermore, let x_{ij}^{kr} and d_i^{kr} be the values of the variables in the point r . Now, by using a binary variable λ^{kr} , the original variables for a certain aircraft k can be expressed as:

$$\begin{aligned} x_{ij}^k &= \sum_{r \in R^k} \lambda^{kr} \bar{x}_{ij}^{kr} \\ d_i^k &= \sum_{r \in R^k} \lambda^{kr} \bar{d}_i^{kr} \\ \sum_{r \in R^k} \lambda^{kr} &= 1 \\ \lambda^{kr} &\in \{0,1\} \quad r \in R^k \end{aligned}$$

Using the results above, FP-MIMF can be written as a Dantzig-Wolfe Master Problem:

Maximize

$$\sum_{k \in A} \sum_{r \in R^k} \left(\sum_{i \in A \cup F} \sum_{j \in F \cup S} c_{ij}^k \bar{x}_{ij}^{kr} - \sum_{j \in F \cup S} c_i \bar{d}_i^{kr} \right) \lambda^{kr} \quad (14)$$

Subject to

$$\sum_{r \in R^k} \lambda^{kr} = 1 \quad k \in A \quad (15)$$

$$\sum_{k \in A} \sum_{r \in R^k} \left(\sum_{j \in F \cup S} \bar{x}_{ij}^{kr} \right) \lambda^{kr} \leq 1 \quad i \in F \quad (16)$$

$$\lambda^{kr} \in \{0,1\} \quad k \in A, r \in R^k \quad (17)$$

To make the model easier to understand and work with, let:

$$a_i^{kr} = 1 \text{ if flight } i \text{ is included in route } r \text{ for aircraft } k, \text{ else } 0. \Rightarrow a_i^{kr} = \sum_{j \in F \cup S} \bar{x}_{ij}^{kr}$$

$$c^{kr} = \sum_{i \in A \cup F} \sum_{j \in F \cup S} c_{ij}^k \bar{x}_{ij}^{kr} - \sum_{i \in F} c_i \bar{d}_i^{kr}$$

Using the substitutions above, the Dantzig-Wolfe Master Problem (DW-MP) is:

Maximize

$$\sum_{k \in A} \sum_{r \in R^k} c^{kr} \lambda^{kr} \quad (18)$$

Subject to

$$\sum_{r \in R^k} \lambda^{kr} = 1 \quad k \in A \quad (19)$$

$$\sum_{k \in A} \sum_{r \in R^k} a_i^{kr} \lambda^{kr} \leq 1 \quad i \in F \quad (20)$$

$$\lambda^{kr} \in \{0,1\} \quad k \in A, r \in R^k \quad (21)$$

The problem is now represented by a set packing model with generalized upper bound (GUB) constraints (19), which ensure that each aircraft is assigned to exactly one route. The second set of constraints (20) ensures that each flight is not included on more than one route. The feasible routes in each set R^k define paths from the aircraft source node to the flight sink node (see Figure 1). Each route may include flights that the aircraft was not originally assigned to, and may also include delayed flights. The costs for the swaps and the delays are subtracted from the revenue that the particular route generates. The objective of the model is to pick one route for each aircraft so that the total revenue is maximized.

3 SOLUTION STRATEGIES

3.1 LP Relaxation

A straightforward approach for solving DW-MP would be to use B&B and thus iteratively solve the LP relaxation of the problem, i.e., DW-LMP. Then all feasible points $R^k \forall k \subseteq A$ have to be generated, i.e., we need to find all paths in the connection network (Figure 1). Even though it is possible to generate them efficiently, the number of feasible paths increases exponentially with the number of flights. Instead, a restricted version of the DW-LMP can be formulated. Let $\mathcal{R}^k \subset R^k$ be a subset of columns for each aircraft. By replacing R^k with $\mathcal{R}^k$ in the model for the DW-MP represented by equations (18) to (21) and perform a linear relaxation, the Dantzig-Wolfe Restricted Linear Master Problem (DW-RLMP) is obtained.

A feasible solution to the DW-RLMP is also a feasible solution to the DW-LMP, and gives a lower bound on the optimal objective function value, i.e., $z^{*}_{DW-RLMP} \leq z^{*}_{DW-LMP}$. The solution also provides dual variables: let π_k and μ_i be the dual variables associated with constraints (19) and (20) respectively. The reduced cost for a new column λ^{kr} is given by:

$$\bar{c}^{kr} = c^{kr} - \pi_k - \sum_{i \in F} \mu_i a_i^{kr} \quad (22)$$

If a new column with a positive reduced cost can be found, it can be added to the DW-RLMP which can be solved once again to provide new dual variables. If no such column can be found, the latest solution to the DW-RLMP is the optimal solution to the DW-LMP. The initial set of columns, $\cup_{k \in A} \mathcal{R}^k$, needs to be constructed so that it is possible to find a feasible solution to the DW-RLMP.

The problem of finding a new column can be formulated as an IP model. As described above, the set of feasible points separates into one set for each aircraft, R^k , which also means one

subproblem for each aircraft. Using the same definitions as for the FP-MIMF but separating the aircraft gives, for example, $x_{ij}^k \rightarrow x_{ij}$, where x_{ij} is equal to one if flight i is followed by flight j . The objective in the subproblem is to maximize the reduced cost for the column, i.e., $\max\{c^{kr} - \pi_k - \sum_{i \in F} \mu_i a_i^{kr}\}$, where a_i^{kr} are used as decision variables. The revenue c^{kr} is a function of a_i^{kr} , but by making the exchange, $a_i^{kr} = \sum_{j \in F \cup S} x_{ij}$, a linear objective function is obtained. The subproblem for finding a new column for aircraft k (DW-SUB^k) can be formulated as:

Maximize

$$\sum_{i \in A \cup F} \sum_{j \in F \cup S} c_{ij}^k x_{ij} - \sum_{i \in F} c_i d_i - \sum_{i \in F} \sum_{j \in F \cup S} \mu_i x_{ij} - \pi_k \quad (23)$$

Subject to

Constraints (2) - (3) and (5) - (12)

where c_{ij}^k and c_i are the same as in FP-MIMF.

If the optimal solution to the problem above has a positive objective function value, a column that may improve the value of the DW-RLMP has been found. The data needed for adding the column to the DW-RLMP is obtained by $a_i^{kr} = \sum_{j \in F \cup S} x_{ij}$ and $c^{kr} = \sum_{i \in A \cup F} \sum_{j \in F \cup S} c_{ij}^k x_{ij} - \sum_{i \in F} c_i d_i$.

The column generation strategy used for solving the DW-LMP can be embedded in a B&B procedure, where new columns are generated for each node of the search tree. In this way a solution to the DW-MP can be obtained. The solution technique is called *Branch and Price* [2] and has proved to be efficient for a large number of problems.

Algorithm 1: Restricted Routefinder

1. Let k be the current aircraft
2. Let $i = k$
3. Let $N = 0$, be a queue of nodes
4. Create a node (the root) for k and add it to N
5. While $N \neq 0$ do
6. Let $M = 0$, $M \leq M_{\max}$, be a set of flights
7. For all $j \in F_i^c \cap F_k^p$
8. Calculate the delay needed if j is to follow i according to equation (6)
9. If the delay is within the bound given by (8) and (11)
10. Calculate the revenue of adding the flight to the path
11. If the revenue is better than the worst revenue for the flights in M
12. Add the flight to M and remove the flight with the lowest revenue from M
13. For all nodes in M
14. Create a new node and add it to N
15. Remove i from N and let i equal the next node in N

16. *Traverse all nodes in the tree*
 17. *If the node can be connected to the flight sink node*
 18. *Create a column by traversing the path from the new node to the root. Calculate the revenue for the path while traversing it and check so that no flight appears twice in the path*
 19. *If the column has a positive reduced cost*
 20. *Add it to DW-RLMP*
-

If the subproblem is solved to optimality for all aircraft, it will return the paths with the highest reduced costs. If the reduced cost for the optimal path is zero, it is clear that an optimal solution to the DW-LMP has been found. Unfortunately, it is a time consuming process to solve the subproblem to optimality.

The subproblem can easily be transformed into a model for the *Shortest Path Problem with Time Windows and Linear Node Costs*. Ioachim et al. [7] present a pseudo polynomial dynamic programming algorithm to solve this problem efficiently but even this efficient algorithm might require more than a second to solve a medium sized subproblem. With one second for solving one sub problem and one subproblem for each aircraft, the time for solving the DW-MP becomes too extended to be of practical value. To solve the subproblem faster, a heuristic can be used. ‘Algorithm 1’ will build a search tree to find paths with positive reduced costs to add to the DW-RLMP.

The revenue for a potential connection in Algorithm 1 (Step 10) is calculated as $c_{ij} - \mu_j - k_{ij}$ where c_{ij} includes the revenue for flying the passengers on flight j as well as the costs for delaying and swapping the flight. μ_j is the dual variable associated with flight j and k_{ij} is a component penalizing the idle time the aircraft spends on the ground. k_{ij} is calculated as the idle time between flight i and flight j multiplied by a constant. The new cost component, k_{ij} , is not used in the total revenue for the path (which is calculated in Step 18), but is helpful in the route generation phase to give incentive for constructing longer routes and accepting flights with delays and swap costs into the route. The maximum number of progeny each node can have is restricted to $Mmax$ (Step 6). This limits the tree size as well as the number of columns that are generated.

If no paths with a positive reduced cost can be found with Algorithm 1, the process terminates and a (not necessarily optimal) solution to the DW-LMP has been found. In many cases, the DW-LMP will yield an integer solution. However, if the solution is fractional, B&B will quickly find an integer solution, provided that no new columns are generated while branching.

3.2 A Lagrangian Heuristic for the DW-MP

Another approach for solving the DW-MP is to use Lagrangian relaxation and subgradient optimization. This procedure has been successfully used for solving the related set covering problem [5]. Instead of relaxing the binary constraints (21) and obtaining the DW-LMP, the packing constraints (20) are moved to the objective function. For any Lagrangian multipliers $\mu_i \geq 0$, the problem DW-MP (μ):

Maximize

$$\sum_{k \in A} \sum_{r \in R^k} c^{kr} \lambda^{kr} - \sum_{i \in F} \sum_{k \in A} \sum_{r \in R^k} (a_i^{kr} \lambda^{kr} - 1) \mu_i \quad (24)$$

Subject to

$$\sum_{r \in R^k} \lambda^{kr} = 1 \quad k \in A \quad (25)$$

$$\lambda^{kr} \in \{0,1\} \quad k \in A, r \in R^k \quad (26)$$

is a relaxation to the DW-MP. For a given set of multipliers, μ_i , the model above describes the *Lagrangian Subproblem* (DW-LS). The *Lagrangian Dual Problem* (DW-LDP) is the problem of finding the optimal multipliers:

$$\begin{aligned} \min h(\mu) = \sum_{i \in F} \mu_i + \quad & \max \sum_{k \in A} \sum_{r \in R^k} (c^{kr} - \sum_{i \in F} a_i^{kr} \mu_i) \lambda^{kr} \\ \text{s.t.} \quad & \sum_{r \in R^k} \lambda^{kr} = 1 \quad k \in A \\ & \lambda^{kr} \in \{0,1\} \quad k \in A, r \in R^k \\ \text{s.t.} \quad & \mu_i \geq 0, \quad i \in F \end{aligned}$$

It is now easy to see that the sub problem separates into one problem for each aircraft. This sub problem can easily be solved by simply picking the column with the highest revenue $c^{kr} - \sum_{i \in F} a_i^{kr} \mu_i$ for each aircraft.

In most cases, the solution to the sub problems does not yield a feasible solution to the set packing model. To obtain one, a greedy heuristic can be used which selects the route with the highest revenue, including the multipliers, that does not duplicate any flights. This is repeated until all aircraft is assigned to one route.

The Lagrangian multipliers are updated with the Polyak subgradient method [11], and the iterative process continues until the gap between the upper bound given by the Lagrangian dual problem and the lower bound given by the best feasible solution found is sufficiently small or the dual objective function value converges. A local search heuristic is also used to improve the feasible solutions, when such are found, using simple swapping techniques.

As for the LP relaxation of the DW-MP, we cannot initially generate all columns, at least not for larger problems. In a similar manner to that described in Section 3.1, the Lagrangian multipliers can be used when generating new columns [6]. The principle is the same as for the column generation scheme based on LP relaxation and is described in ‘Algorithm 2’. In each iteration a restricted version of the master problem, DW-RMP, is solved to optimality or near optimality.

Algorithm 2: Lagrangian Column Generation

1. Let $\tau = 0$ be an iteration counter
2. Let ε be a sufficiently small number
3. Initiate the multipliers, i.e., $\mu_i^\tau = 0 \quad \forall i$
4. Generate $Y^k \subset R^k \quad \forall k \in A$
5. Repeat
6. $\tau = \tau + 1$
7. $\mu_i^\tau = \mu_i^{\tau-1}$
8. Solve the DW-RMP with $Y^k \quad \forall k \in A$, using the Lagrangian heuristic

9. Let μ_i^τ be the optimal multipliers and $c_k^\tau = c^{kr^*} - \sum_{i \in F} a_i^{kr^*} \mu_i^\tau$ the revenue for the optimal column, r^* , for aircraft k
 10. For all k , generate new columns, r , whose revenue $c^{kr} - \sum_{i \in F} a_i^{kr} \mu_i^\tau$ is higher than c_k^τ and add them to Y^k
 11. Until $\sum_{i \in F} |\mu_i^\tau - \mu_i^{\tau-1}| < \varepsilon$
-

The main difference from column generation based on LP relaxation is when to decide whether a column should be added to the problem. In the LP case, the reduced cost for a new column can be calculated, but this is not directly possible when Lagrangian relaxation is used. However, when the DW-LS is solved, the best column as calculated by $c^{kr} - \sum_{i \in F} a_i^{kr} \mu_i^\tau$ is picked for each aircraft. In Step 9, the revenue c_k^τ is the revenue for the best column that can be found with the optimal multipliers μ^τ . If a column with higher revenue could be found, that would be picked instead when the subproblem is solved. Therefore any new column with such revenue will improve the solution of the DW-LS and can be added to the DW-RMP.

In Step 10, new columns are generated by Algorithm 1, which can be used as described earlier but with minor alterations. Instead of checking if a column has a positive reduced cost (Step 19), the check done in Step 10 in Algorithm 2 is performed. The stop condition in Step 11 is only one possible choice. Another would simply be to stop when no new columns can be found in Step 10.

4 COMPUTATIONAL RESULTS

The purpose of providing computational results is not only to show the efficiency of the models and algorithms that have been developed. It is also to show how a flight controller actively could use this kind of decision support tool to obtain a number of different suggestions for solutions.

In Section 4.1 below, all columns are generated before the DW-MP is solved with the Lagrangian heuristic. This is to provide an upper bound on the solutions and to show how the solution time increases beyond practical use in the larger problems. In this section, the way a flight controller may use the weights to adjust the solutions is also described. The results for the column generation strategies are accounted for in Section 4.2, but first of all the data sets and the technical conditions are described below.

Two different data sets, $s1$ and $s2$, are used to evaluate the solution strategies. Both sets are obtained from a Swedish domestic airline, which means that the flights are short, between 15 and 125 minutes. Table 1 sums up the important characteristics for the data sets.

	Data Set	
	$s1$	$s2$
Number of aircraft	13	30
Number of aircraft types	2	5
Number of flights	98	215
Number of airports	19	32

Table 1: The two data sets

The two data sets are perturbed in two different ways, a and b , where a is an aircraft breakdown and b the imposed delay on a number of flights in the set. This gives the data sets $s1a$, $s1b$, $s2a$, and $s2b$. In $s1a$ an aircraft is unavailable for 5 hours, and in $s1b$ two flights are imposed with delays for 25 and 30 minutes respectively. In $s2a$ an aircraft is unavailable for six hours, while in $s2b$ four flights are imposed with delays ranging from 15 up to 40 minutes. A required turnaround time of 10 minutes is used for all flights and aircraft. The maximum delay allowed for a single flight, D_i , is set to 60 minutes for all flights and a feasible route may contain a maximum of 16 flights.

Each data set is solved using different weights to calculate the costs. This is to show how a dispatcher would be able to change the weights if the solution obtained does not meet the requirements, as described in Section 2.1.

[All solution strategies are implemented as C++ programs. The code is compiled with the Sun WorkShop Compiler C++ 5.0 and the tests are performed on a Sun Enterprise 450 computer with 1024 Mb of memory and four 400 MHz processors, running under Solaris 2.7. No parallel computing is performed though, so only one processor is utilized in each run. ILOG CPLEX 7.0 is used for solving the LP problems in the LP relaxation strategy, while no commercial optimization software is used in the Lagrangian heuristic.]

4.1 Prior Generation of Columns

Generating all pertinent columns for the data sets takes some time; about one second for the smaller data sets ($s1a$ and $s1b$) and about a minute for the larger ones ($s2a$ and $s2b$).

n	D-set	Weights				Results							
		can	sw-t	sw	del	t	z	UDB	Gap[%]	c	st	s	d
1	$s1a$	20	100	10	1	4	43840	43844	0.009	30	2	8	0
2	$s1a$	20	1000	10	1	4	43760	43761	0.002	46	0	4	0
3	$s1a$	100	1000	10	1	6	218980	218980	0	32	0	0	1420
4	$s1a$	100	100	10	1	9	220680	220701	0.010	25	4	2	0
5	$s1b$	20	100	10	1	5	44600	44600	0	0	0	12	0
6	$s1b$	20	100	100	1	2	44070	44070	0	0	0	4	250
7	$s1b$	20	100	400	1	1	43140	43140	0	0	0	0	1580
8	$s2a$	20	100	10	1	645	68480	68487	0.010	29	2	4	0
9	$s2a$	20	1000	10	1	526	68000	68005	0.007	65	0	0	0
10	$s2a$	100	1000	10	1	1139	341865	3411871	0.002	29	0	12	1615
11	$s2b$	20	100	10	1	1118	69080	69086	0.009	0	0	12	100
12	$s2b$	20	100	10	10	1050	68720	68726	0.009	23	0	12	0
13	$s2b$	20	100	50	1	583	68900	68907	0.010	0	2	2	100

Table 2: Results for the Lagrangian heuristic

The results for the Lagrangian heuristic are presented in Table 2. The first column, n , contains a run label, so that the individual runs can be easily referred to. The column D -set notes which data set is used while the *Weights* are the ones used for calculating the costs in the model (see Section 2.1). *can* is the weight for cancelling a flight, *sw-t* the weight for assigning a flight to an aircraft of a different type, *sw* the weight for assigning a flight to an aircraft in the same fleet as originally planned, and *del* is the weight for delaying a flight.

In the *Result* section, t is the solution time in seconds and z the objective function value for the best solution found. UBD is an the upper bound on the optimal solution that is provided by the heuristic and Gap is the gap (as a percentage) between the solution found and the UBD . c , st , s and d stand for cancellations, swaps between aircraft types, regular swaps and delay, i.e., the characteristics of the best solution found. c notes how many passengers that are affected by a canceled flight, st and s how many swaps between aircraft types and regular swaps that are made, and finally d is the total amount of passenger delay in the solution.

The four different problems provided by the four data sets have been solved with a number of different weights. These have been set so that each setting produces a unique solution with a different characteristic than the other solutions. For example, in run 1 the solution includes 30 canceled passengers, 2 swaps between different aircraft types, 8 regular swaps and no delayed flights.

Now, if the flight controller does not like swaps between aircraft types, the corresponding weight may be increased from 100 to 1000, which gives the setting in run 2. In run 2 the solution includes no swaps between aircraft types, but instead the number of passengers affected by a canceled flights has increased to 46, which might be too much for the flight controller who thereby raises the cancellation weight to 100 from 20. Thus in run 3, the number of canceled passengers has decreased to 32, at the expense of a number of delayed flights that give a total passenger delay of 1420. Going back to setting 1 and accepting swaps between aircraft types, a user may instead increase the weight for cancellations to obtain the setting in 4 and find a solution with even less canceled passengers at the expense of an increasing number of swaps between aircraft types.

Looking at data set *s1b*, two flights have been imposed with shorter delays. Therefore the intuitive solution to the problem would be to delay flights, but as can be seen in run 5, it is possible to reassign flights, i.e., performing swaps, so that no flight has to be delayed. If this solution contains too many swaps, the corresponding weight can be raised from 10 to 100, and the solution in run 6 obtained where the number of swaps has decreased from 12 to 4 but where there also is a total delay of 250 passenger minutes. By increasing the swap weight even further to 400 and obtaining the setting for run 7, the solution will not include any swaps at all, but instead only delayed flights.

The data set *s2a* has an aircraft shortage for a couple of hours and the weight setting in run 8 gives a solution with 29 canceled passengers, 2 swaps between aircraft types, and 4 regular swaps to solve this problem. If the weight for swaps between aircraft types is increased, as in run 9, all perturbed flights are canceled, which affects 65 passengers. By increasing the cancellation weight as well, the same flights as in the first run are canceled, but instead of fleet swaps, more regular swaps are performed and some flights are delayed.

In the last data set, a number of flights are imposed with delays. The solution in run 11 uses 12 swaps and also delays some flights. By increasing the delay weight, the setting in run 12 is obtained, where some passengers are affected by cancellations and the number of swaps increases. The final run shows that it is also possible to solve the problem with only 2 regular swaps if 2 swaps between aircraft types are accepted.

The largest gap between the best solution found and the upper bound for any of the runs is 0.01 percent, which must be considered as quite small. Looking at the solution times, the strategy appears to be quite effective for the smaller data sets, even including the prior-generation of columns. However, for the data sets *s2a* and *s2b*, the time a flight controller would have to wait for a solution is simply too long, in the worst case almost 20 minutes, to be of practical value. It is therefore interesting to see how well the more heuristic column generation approaches perform.

4.2 The Column Generation Strategies

The column generation strategies use Algorithm 1 to generate new columns. When no new columns can be found, the maximum number of progeny nodes allowed is increased by one. The strategies are run for 60 seconds and the best feasible solution found (if any) is reported every five seconds. The strategies will of course show a different performance depending on how many progeny nodes are allowed from the beginning of the solution process [initial testing has shown that three progeny nodes works well].

In the LP case, only the linear relaxation of the DW-MP is solved, which means that some of the solutions reported may be fractional and therefore not directly usable. It is, however, easy to redefine the variables as integer and solve the corresponding IP problem with the columns that have already been generated, although this is not done here.

n	Strat	Results (% from BKS)				
		5 s	10 s	15 s	30s	60 s
1	LH	0	0	0	0	0
	LP	0	0	0	0	0
2	LH	0	0	0	0	0
	LP	0	0	0	0	0
3	LH	0	0	0	0	0
	LP	0	0	0	0	0
4	LH	0	0	0	0	0
	LP	0.009	0	0	0	0
5	LH	0	0	0	0	0
	LP	0	0	0	0	0
6	LH	0	0	0	0	0
	LP	0	0	0	0	0
7	LH	0	0	0	0	0
	LP	0	0	0	0	0
8	LH	0.701	0.146	0.146	0.146	0.146
	LP	0.803	0.803	0.234	0.058	0.058
9	LH	0	0	0	0	0
	LP	0	0	0	0	0
10	LH	0.259	0.259	0.259	0.259	0.165
	LP	0.342	0.229	0.191	0.107	0.085
11	LH	0.615	0.615	0.413	0.413	0.232
	LP	0.615	0.290	0.290	0.290	0.145
12	LH	0.349	0.349	0.291	0.291	0
	LP	0.349	0.349	0.349	0.349	0.058
13	LH	0.356	0.356	0.356	0.356	0.239
	LP	0.356	0.356	0.356	0.356	0.239

Table 3: Results for the column generation heuristics

The results for the column generation strategies are presented in Table 3. The data sets and the weights are the same as used in the runs for the Lagrangian heuristic, accounted for in Table 2, e.g., in run 5, data set *s1b* is used with a cancellation weight of 20, a weight for swaps between different aircraft types of 100, regular swaps 10 and the delay weight set to 1.

In the *Strat* column, LH means that the column generation strategy based on the Lagrangian heuristic has been used while LP means that the LP based strategy was used. The results are presented in percentage terms from the best solution found by the Lagrangian heuristic where the routes had been generated beforehand, i.e., the results presented in Table 2. The results are given when the strategies have been run for 5, 10, 15, 30 and 60 seconds. For example, in run 11, the best solution after 10 seconds produced by the Lagrangian column generation heuristic (LH) had an objective value of 68655. The best solution found for this data set and weight setting is worth 69080 according to Table 2, and $(69080-68655)/69080 \approx 0.00615$.

Considering runs 1 - 7, where the smaller data sets were used, the zeros are dominating, indicating that the strategies found the best known solution; the BKS was always produced within 10 seconds.

The larger data sets are accounted for in runs 8 - 13, and here the strategies often fail to find the best known solution. However, the solutions found are always less than one percent from the best known. The worst result produced by any of the strategies was 0.803 % from the best known solution (run 8, the LP strategy after 5 seconds). After 30 seconds this result has improved to 0.058 %. It is hard to draw conclusions as to which of the two strategies, LH or LP, has the better performance. Both produce high quality solutions quickly and since the only difference between them is the way of solving the master problem, a similar performance can be expected. The results indicate that the LH strategy might be slightly better when a short solution time is desired (runs 4, 8 and 10, 5 seconds) while the LP strategy produces better solutions when it is allowed to run for a longer time (runs 8, 10 and 11, 60 seconds). However, it is important to remember that there is no guarantee that the solutions presented by the LP strategy are integer, even if this is mostly the case.

5 CONCLUSIONS AND POSSIBLE EXTENSIONS

The computational results show that the method proposed in this paper for solving the FPP is efficient and can provide a flight dispatcher with a number of different suggestions. The users can state their preferences in the form of weights, which are used to calculate the costs in the model, and in this way affect the number of canceled flights, delayed flights and swaps performed in the solution.

The column generation heuristics can handle larger problem instances and produce good solutions to the problem. However, the solution time and quality could still be improved and one possible way of doing this is to find a better way than Algorithm 1 of generating routes.

One of the prime benefits with the proposed model and solution strategy is that it is easy to incorporate additional constraints when constructing feasible routes for the aircraft. In this paper, that quality has not been exploited. A logical extension to the research presented here is therefore to consider more of the characteristics, such as e.g., maintenance considerations, crew constraints or slot times, which may cause a solution to be infeasible.

Column based models, such as set partitioning and set covering, have been extensively used when solving crew scheduling problems. A column in such a model typically represents a feasible pairing, which is a sequence of flight legs that does not violate the numerous air traffic regulations and rules in the collective agreements. A solution to the crew pairing problem consists of a legal set of minimum cost pairings that cover all flight legs in the schedule. There is no guarantee that a solution to the FPP will not violate some of the crew

related rules. However, the problem of constructing feasible pairings has been extensively studied and there exist efficient algorithms for solving it. It should be possible to merge such an algorithm with the route construction algorithms presented here, and thus obtaining feasible routes to the FPP that do not violate any crew constraints.

Note and Acknowledgments

This research was partly sponsored by RM Rocado. The authors would like to thank RM Rocado for the opportunity of working with this problem and for supplying real problem data. RM Rocado is a Swedish company that develops planning systems for airlines. In order to develop an improved decision support system for the operational planning of aircraft, RM Rocado turned to the University of Linköping. The application they wanted to develop was a fast dynamic decision support system that in a few seconds was able to suggest revised aircraft schedules when a perturbation had made it impossible to use the original one.

References

- [1] M. Arguello, J. Bard, G. Yu, A GRASP for aircraft routing in response to groundings and delays. *Journal of Combinatorial Optimization*, **1** (3), 211-228 (1997).
- [2] C. Barnhart, E. Johnson, G. Nemhauser, M. Savelsbergh, P. Vance, Branch-and-Price: Column generation for solving huge integer programs. *Operations Research*, **46** (3), 316-329 (1998).
- [3] J. Cao, A. Kanafani, Real-time decision support for integration of airline flight cancellations and delays part 1: Mathematical formulation. *Transportation Planning and Technology*, **20**, 183-199 (1997).
- [4] J. Cao, A. Kanafani, Real-time decision support for integration of airline flight cancellations and delays part 2: Algorithm and computational experiments. *Transportation Planning and Technology*, **20**, 201-217 (1997).
- [5] A. Caprara, M. Fischetti, P. Toth, A heuristic method for the set covering problem. *Operations Research*, **47**, 730-743 (1999).
- [6] M. Henningsson, *Some optimization models in the area of telecommunications*. Licentiate Thesis, Division of Optimization, Department of Mathematics, Linköping University, Linköping, Sweden, 1998.
- [7] I. Ioachin, S. Gélinas, F. Soumis, J. Desrosier, A dynamic programming algorithm for the shortest path problem with time windows and linear node costs. *Networks*, **31**, 193-204 (1998).
- [8] A. I. Jarrah, G. Yu, N. Krishnamurthy, A. Rakhit, A decision support framework for airline flight cancellations and delays. *Transportation Science*, **27**, 266-280 (1993).
- [9] M. Love, K. Sorensen, J. Larsen, J. Clausen, *Using heuristics to solve the dedicated aircraft recovery problem*. Technical Report IMM-TR-2001-18, Department of Informatics and Mathematical Modelling, Technical University of Denmark, Kgs. Lyngby (2001).
- [10] D. F. X. Mathaisel, Decision support for airline system operations control and irregular operations. *Computers and Operations Research*, **23**, 1083-1098 (1996).
- [11] B. T. Polyak, Minimization of unsmooth functionals. *USSR Computational Mathematics and Mathematical Physics*, **9**, 14-29 (1969).
- [12] A. Rakshit, N. Krishnamurthy, G. Yu, System operations advisor: A real-time decision support system for managing airline operations at United Airlines. *Interfaces*, **26** (2), 50-58 (1996).

- [13] D. Teodorovic, S. Guberinic, Optimal dispatching strategy on an airline network after a schedule. *Journal of Operations Research*, **15**, 178-182 (1984).
- [14] D. Teodorovic, G. Stojkovic, Model for operational daily airline scheduling. *Transportation Planning and Technology*, **14**, 273-285 (1990).
- [15] D. Teodorovic, G. Stojkovic, Model to reduce airline schedule disturbances. *Journal of Transportation Engineering*, **121** (4), 324-331 (1995).
- [16] B. Thengvall, J. Bard, G. Yu, Balancing user preferences for aircraft schedule recovery during irregular operations. *IIE Transactions*, **32** (3), 181-193 (2000).
- [17] B. Thengvall, G. Yu, J. Bard, Multiple fleet aircraft schedule recovery following hub closures. *Transportation Research A*, **35**, 289-308 (2001).
- [18] S. Yan, D. H. Yang, A decision support framework for handling schedule perturbation. *Transportations Research B*, **30** (6), 405-419 (1996).
- [19] S. Yan, Y. Tu, Multifleet routing and multistop flight scheduling for schedule perturbation. *European Journal of Operations Research*, **103**, 155-169 (1997).
- [20] S. Yan, C. G. Lin, Airline scheduling for the temporary closure of airports. *Transportation Science* **31** (1), 72-82 (1997).