



An iterative approach to robust and integrated aircraft routing and crew scheduling

Oliver Weide^{a,*}, David Ryan^a, Matthias Ehrgott^{a,b}

^aDepartment of Engineering Science, The University of Auckland, New Zealand

^bLaboratoire d'Informatique de Nantes Atlantique, Université de Nantes, France

ARTICLE INFO

Available online 5 April 2009

Keywords:

Airline scheduling
Crew pairing
Aircraft routing
Robust

ABSTRACT

In airline scheduling a variety of planning and operational decision problems have to be solved. We consider the problems aircraft routing and crew pairing: aircraft and crew must be allocated to flights in a schedule in a minimal cost way. Although these problems are not independent, they are usually formulated as independent mathematical optimisation models and solved sequentially. This approach might lead to a suboptimal allocation of aircraft and crew, since a solution of one of the problems may restrict the set of feasible solutions of the problem solved later. Also, when minimal cost solutions are used in operations, a short delay of one flight can cause very severe disruptions of the schedule later in the day. We generate solutions that incur small costs and are also robust to typical stochastic variability in airline operations. We solve the two original problems iteratively. Starting from a minimal cost solution, we produce a series of solutions which are increasingly robust. Using data from domestic airline schedules we evaluate the benefits of the approach as well as the trade-off between cost and robustness. We extend our approach considering the aircraft routing problem together with two crew pairing problems, one for technical crew and one for flight attendants.

© 2009 Elsevier Ltd. All rights reserved.

1. Introduction

A sequence of planning problems must be solved in airline scheduling: first, marketing decisions in the *schedule design problem* determine the schedule of flights the airline operates. Each flight is specified by origin, destination, departure date, departure time and duration. Given the set of flights in a schedule, the solution of the *fleet assignment model* determines which flight is operated by which aircraft type. The objective is to maximise profit with respect to the number of available aircraft and other resource constraints. Next, the *aircraft routing problem* seeks a minimal cost assignment of available aircraft to the flights. A routing is assigned to each individual aircraft such that each flight is covered by exactly one routing. The routings must satisfy maintenance restrictions. Once aircraft types are assigned to flights, the aircraft routing problem can be solved for each aircraft type separately. In a similar way to the aircraft routing problem, the *crew pairing problem* (or *tour of duty problem*) allocates crew to flights in a minimal cost way. A set of generic crew pairings is constructed subject to many rules so that each flight is covered exactly once. Under the assumption that the crew is only allowed to operate a single aircraft type (which is usually the case for pilots, for example) the crew pairing problem

can also be solved separately for each aircraft type. The last of the planning problems is *crew rostering*. Based on the constructed crew pairings, a line of work is assigned to each individual crew member. The reader is referred to Klabjan [18] for a detailed description of the various airline scheduling problems.

In this paper two of the above problems, aircraft routing and crew pairing, are considered. The fleet assignment model is important for large airlines with multiple aircraft types. In the context relevant for this paper, the fleet can be regarded as homogeneous and fleet assignment can be omitted. The crew rostering problem can be viewed as a separate optimisation problem with no influence on the cost of the overall solution and is also not considered.

Traditionally, the aircraft routing problem is solved prior to the crew pairing problem, although the two problems are interdependent. One of the important causes of delays in airline operations is the availability of crew. This is particularly true if the crew change from one aircraft to another during a duty period, especially if there is minimum ground time between the flights. If the incoming flight has been delayed for some reason, then both that aircraft and the aircraft to which the crew are changing will depart late. This propagation of delay can cause serious disruptions to the operation of the flight schedule. Ehrgott and Ryan [10] and Yen and Birge [36] have shown that the “robustness” of crew pairing solutions can be significantly improved if aircraft changes (AC) are made when ground time between the incoming and the outgoing flights is much greater than the minimum ground time. This can be achieved in the crew pairing problem by penalising aircraft changes when ground time is

* Corresponding author.

E-mail addresses: o.weide@auckland.ac.nz (O. Weide), d.ryan@auckland.ac.nz (D. Ryan), m.ehrgott@auckland.ac.nz (M. Ehrgott).

short. Robust crew pairing solutions then have “crew following the same aircraft” as much as possible and changing aircraft only when ground time between flights is much longer than the minimum. In this sense, the robust crew pairing solution depends on the given aircraft routing solution. It is natural to ask if it is possible to improve the robustness of the crew pairing solution by permitting changes to the aircraft routing solution to encourage the “aircraft to follow the crew”. In this paper we investigate this question and show that it is indeed possible to reduce the cost of the crew pairing solution and simultaneously increase its robustness by considering the two problems together.

We formulate the aircraft routing problem and the crew pairing problem in one integrated model. This model yields one optimal solution for the two problems where the objective function is a weighted sum of cost and some value attached to robustness. However, the model is very hard to solve. Integration increases the complexity of both problems which are already \mathcal{NP} -hard individually (Garey and Johnson [13]). Decomposition methods are proposed in the literature to solve the integrated problem (see Mercier et al. [24]) but long computation times are needed to solve the model to optimality.

Instead of solving the integrated model, we propose to solve the two original problems iteratively. We start with a minimal cost crew pairing solution without taking aircraft routings into account. Then, in each iteration we solve the individual aircraft routing problem first, taking into account the current crew pairing solution. Then, given the aircraft routing solution we resolve the crew pairing problem. We only use the objective functions in both problems to pass information from the problem solved previously to generate more and more robust solutions. Hence the constraints are unaltered and the complexity of the two problems is not increased. We stop the process when the level of robustness cannot be improved any further.

This procedure generates a series of feasible solutions for the integrated model with varying cost and robustness measure. The airline is not required to associate a cost with robustness but can study the trade-off between cost and robustness and then choose a solution they prefer to operate.

Applying this approach to various domestic airline schedules we obtain low cost solutions that are highly robust. The solutions obey all rules imposed by the airline and are ready to be implemented in practice. Various crew groups such as captains, first officers and flight attendants are required to operate an aircraft. We therefore extend the iterative approach and consider multiple crew groups at the same time. We generate solutions for the aircraft routing problem together with solutions for two different crew groups and compare these with solutions generated by applying the iterative approach to each crew group separately.

The paper is organised as follows: In Section 2 the problems are formally stated. We review the most recent approaches in the literature in Section 3. In Section 4 the iterative solution approach is presented and Section 5 concludes with numerical results.

2. Problem formulation

In this section we define the crew pairing problem and the aircraft routing problem. We discuss how both problems can be solved efficiently and we formulate a model that integrates the two problems. The integrated model is capable of generating solutions that incur small costs and are also operationally robust. We start with the crew pairing problem.

2.1. The crew pairing problem

Given a flight schedule, the *crew pairing problem* is defined as the problem of assigning crews to flights in the schedule such that each

flight is operated by exactly one crew. A sequence of flights which can be flown by a crew on one work day is called a *duty period*. An alternating sequence of duty periods and rest periods is called a *crew pairing* (or *tour of duty*). Any crew pairing must start and end at the same crew base and is restricted by a number of rules such as rest time regulations or flying time restrictions. Costs are associated with each crew pairing. In the crew pairing problem we seek a minimal cost set of pairings that partition the flights in the schedule, i.e. each flight is contained in exactly one pairing.

The pairings can be represented as columns of a binary $m \times n^p$ matrix A^p where m is the number of flights in the schedule and n^p is the number of possible pairings. Entry $(a_{ij})^p$, $1 \leq i \leq m$, $1 \leq j \leq n^p$ equals 1 if flight i is contained in pairing j and 0 otherwise. With this matrix representation we formulate the crew pairing problem as a standard *set partitioning model* (Wolsey [35]):

$$\begin{aligned} & \text{Minimise} && (\mathbf{c}^p)^T \mathbf{x}^p \\ & \text{subject to} && A^p \mathbf{x}^p = \mathbb{1} \\ & && \mathbf{x}^p \in \{0, 1\}^{n^p}. \end{aligned} \quad (1)$$

The element c_j^p of $\mathbf{c}^p \in \mathbb{R}^{n^p}$ is the cost associated with pairing j . The decision variable $x_j^p \in \{0, 1\}$ is equal to 1 if pairing j is contained in the solution and 0 otherwise. The cost of a pairing is composed of flight time and duty time salaries, and meal, rest and travel allowances. *Base-constraints* are added to the standard model to consider *base strengths* at the crew bases. The base strength restricts the number of crew pairings that can start at a crew base in a particular week or on a particular day. With base-constraints included the model is referred to as *generalised set partitioning model* since these constraints usually have non-unit right hand sides.

Because the number of pairings n^p is too large to efficiently generate all of them, column generation and branch-and-bound techniques (i.e. branch-and-price) are identified in the literature as the most successful methods to solve the problem (Barnhart et al. [1]), which is \mathcal{NP} -hard. Using *column generation* only a small fraction of all possible pairings is considered initially and additional pairings (columns) are generated during the execution of the algorithm.

We model the schedule as a directed *flight network* (Minoux [25]) where arcs represent flights and nodes represent departure or arrival of a flight. Additionally, a *connection-arc* is linking the arrival of one flight with the departure of another flight if the two flights can be operated consecutively by the same crew. In this network each crew pairing is represented by a path and the column generation problem can be solved by a resource constrained shortest path problem. The rules the pairing must obey are incorporated into the shortest path algorithm as resource constraints. A second network type common in the literature is the *duty period network*. This network consists of an arc for each possible duty period and overnight arcs linking the duty periods (Lavoie et al. [21]).

To solve (1), first the linear program (LP) relaxation of (1) is solved using column generation. Fractions appearing in this solution are caused by different crew pairings competing for the same flights. To eliminate these fractions and obtain an integer solution for (1), a branch-and-bound algorithm with a *constraint branching* strategy is used (Ryan and Foster [29]). In this strategy two flights in the solution are chosen that are partially covered by different crew pairings. In one branch solutions are considered where both flights are contained in the same crew pairing. In the other branch both flights must not be operated by the same crew pairing. A special case of constraint branching is *follow-on branching* (Ryan and Falkner [28]) where two flights must be operated consecutively by the same crew in one branch and must not be operated consecutively by the same crew in the other branch.

Other approaches to solve the crew pairing problem include Vance et al. [34] and Desaulniers et al. [9]. Vance et al. [34]

decomposed the problem into two stages, first partitioning the flights by duty periods and then the duty periods by pairings. Desaulniers et al. [9] used a nonlinear multi-commodity network flow formulation and solved it with a Dantzig–Wolfe decomposition method.

2.2. The aircraft routing problem

The *aircraft routing problem* (also referred to as *tail assignment*) is the problem of assigning aircraft to a given set of flights in a schedule. We seek one routing for each aircraft such that each flight of the schedule is contained in exactly one routing. Each routing is subject to maintenance requirements and other flying restrictions and the number of available aircraft is given. Each particular aircraft must be assigned to one specific routing, similar to the *crew rostering problem* (Day and Ryan [8], Kohl and Karisch [20]) where a line of work is assigned to a particular crew member.

In a similar way to the crew pairing problem, routings can be represented as columns of a binary $(m + a) \times n^R$ matrix A^R where m is the number of flights, a the number of available aircraft and n^R the number of possible routings. The first m rows are defined similarly to those in A^P : the element $(a_{ij})^R, 1 \leq i \leq m, 1 \leq j \leq n^R$ equals 1 if flight i is contained in routing j and 0 otherwise. Additionally, the element $(a_{m+i,j})^R, 1 \leq i \leq a, 1 \leq j \leq n^R$ equals 1 if routing j is operated by aircraft i and 0 otherwise. The last a constraints are referred to as *generalised upper bound (GUB)* constraints (or aircraft convexity constraints) and ensure that each aircraft is assigned to exactly one routing. With this matrix representation the aircraft routing problem can be formulated in a similar manner to the crew pairing problem:

$$\begin{aligned} & \text{Minimise } (\mathbf{c}^R)^T \mathbf{x}^R \\ & \text{subject to } A^R \mathbf{x}^R = \mathbb{1} \\ & \mathbf{x}^R \in \{0, 1\}^{n^R}. \end{aligned} \tag{2}$$

The element c_j^R of $\mathbf{c}^R \in \mathbb{R}^{n^R}$ is the cost associated with routing j . The decision variable $x_j^R \in \{0, 1\}$ is equal to 1 if routing j is in the solution and 0 otherwise. As costs of a routing *through-values* are usually considered. A through-value is revenue attached to a pair of flights if they are operated in sequence by the same aircraft. This accounts for passenger demand for direct connections (i.e. without the necessity of changing aircraft) on certain itineraries. In the context of this paper a set of connections that must be operated by the same aircraft is given by the airline and hence no through-values are considered. Since we also assume that the operational cost of all aircrafts are identical, the aircraft routing problem reduces to a feasibility problem. We introduce penalty costs for the aircraft routings to influence characteristics of the aircraft routing solution, see Section 4.

This form of the set partitioning model is called the *rostering model* and is also \mathcal{NP} -hard. As in the crew pairing problem, the number of possible routings is very large and column generation techniques are used to obtain a solution for the problem. Using the same network formulation as the crew pairing problem, aircraft routings can also be represented as paths. The column generation problem is formulated as a resource constrained shortest path problem where we again only generate columns which satisfy all maintenance rules.

In contrast to the crew pairing problem we now branch on aircraft–flight pairs to obtain integer solutions. In one branch a particular aircraft is forced to operate a particular flight while in the other branch the aircraft is not allowed to operate this flight.

To justify this strategy we refer to the theory of *perfect matrices* which was first proposed by Padberg [26]. For a perfect matrix A and the problem $\min\{\mathbf{c}^T \mathbf{x} : A\mathbf{x} = \mathbb{1}, \mathbf{x} \geq 0, \mathbf{c} \in \mathbb{Z}^n\}$ there always exists an optimal integral solution vector \mathbf{x} . We first introduce some notation:

A graph G is called *complete* if every node is adjacent to every other node. The *chromatic number* of G is the minimal number of

different colours needed to colour all nodes of G such that no adjacent nodes have the same colour. A *subgraph* G' of G is a subset of the nodes of G together with all arcs of G linking nodes in the subset. A *clique* is a complete subgraph. A graph G is called *perfect* if for every subgraph G' of G the chromatic number of G' is equal to the maximal cardinality of a clique in G' .

Let G_i denote the *intersection graph* associated with a matrix A . The nodes of G_i correspond to columns of A and two nodes are linked by an edge if the two corresponding columns have a common 1 in any row. The matrix A is called *perfect* if the associated intersection graph is perfect.

We investigate the submatrices of A^R that consist of the columns of A^R associated with a single aircraft. The intersection graphs of these submatrices are complete since all columns have a common 1 in the aircraft convexity constraint. Every subgraph of a complete graph is also complete. Also, in any complete graph the chromatic number equals the cardinality of a maximal clique which is equal to the number of nodes in the graph. This results in the following theorem:

Theorem 1. *Each submatrix of A^R , which consists of the columns associated with a single aircraft, is perfect.*

Hence, fractions in the solutions of the LP relaxation of the aircraft routing problem are caused by different aircraft competing for the same flight.

Like for the crew pairing problem alternative solution methods for the aircraft routing problem include multi-commodity network flow formulations (e.g. Cordeau et al. [6]).

2.3. Robust and integrated crew pairing and aircraft routing problem

If two flights can be operated in sequence by the same crew or aircraft (i.e. there exists a connection–arc linking both flights), the time between arrival of the first and departure of the second flight is called *turn-time* for aircraft and *sit-time* for crew. In the context of the paper arrival and departure of a flight refer to arrival at and departure from the gate, respectively.

The minimal time required for an aircraft or crew to operate a connection is called *minimal turn-time* or *minimal sit-time*, respectively. The minimal sit-time is always greater or equal to the minimal turn-time. If the crew stays on the same aircraft the minimal turn-time of the connection applies for the crew instead of the minimal sit-time. A connection between flights i and j is called *short* if

$$(\text{minimal turn-time})_{ij} \leq (\text{sit-time})_{ij} < (\text{minimal sit-time})_{ij},$$

where the first part of the inequality holds for any feasible connection. Thus, in a feasible solution, short connections are only allowed if the crew stays on the same aircraft. If the two problems are solved separately this condition might result in suboptimal (in case the aircraft routing problem is solved first) or infeasible solutions (in case the crew pairing problem is solved first).

Also, we prefer solutions where crew are not changing aircraft when the sit-time is less than some *restricted time*. A connection between two flights i and j is called *restricted* if

$$(\text{minimal sit-time})_{ij} \leq (\text{sit-time})_{ij} < (\text{restricted time})_{ij}.$$

The concept of restricted connections is first introduced by Mercier et al. [24]. In contrast to short connections, crews are allowed to change aircraft if the connection is restricted, but we try to find solutions in which this occurs as rarely as possible. If crew change aircraft on restricted connections we refer to these connections as *restricted aircraft changes (RAC)*.

It is unlikely that the schedule is operated as planned. Delays occur frequently in airline operations and can, for example, be caused

by late passengers, unscheduled maintenance requirements or bad weather. Small turn-times are usually used in aircraft routings to keep costs low and connection times attractive for passengers. Hence, if a flight is delayed, the flight operated next by the aircraft is probably also delayed. But if the crew are also changing aircraft on a restricted connection after the delayed flight, at least one more flight might be affected by the initial delay. Due to the small buffer to compensate for the delay, the crew are likely to be late for the next flight they operate. This behaviour can propagate to a large number of delayed flights in a short amount of time.

A solution where effects of potential delays are minimal is called *operationally robust*. The concept of robust solutions is important since an airline is interested in achieving high *on-time performance (OTP)*, i.e. the percentage of all flights in the schedule that depart on-time. Bad OTP can incur large additional costs (referred to as *recovery costs*), caused by additionally required crews, compensations for passengers affected by delayed or cancelled flights and damaged reputation of the airline. These costs may by far exceed the savings of using a solution with less *planned cost* than using a solution that is more robust. We try to identify solutions with low planned costs which are operationally robust, i.e. where disruptions will result in minimal recovery costs. Costs listed in this paper are generally planned costs. We refer to the sum of planned costs and recovery costs as *operational costs*.

In order to integrate the concepts of short and restricted connections into our formulation we enumerate all possible short and restricted connections. We define a binary $m^B \times n^P$ matrix B^P where m^B is the number of short connections. Each pairing is associated with one column of B^P , where $(b_{ij})^P, 1 \leq i \leq m^B, 1 \leq j \leq n^P$ equals 1 if short connection i is contained in pairing j and 0 otherwise. Analogously for restricted connections, we define a binary $m^D \times n^P$ matrix D^P where m^D is the number of restricted connections. $(d_{ij})^P, 1 \leq i \leq m^D, 1 \leq j \leq n^P$ equals 1 if restricted connection i is contained in pairing j and 0 otherwise. For aircraft a binary $m^B \times n^R$ matrix B^R and a binary $m^D \times n^R$ matrix D^R are defined in an analogous way.

With this matrix representation the *robust and integrated crew scheduling and aircraft routing problem* can be formulated as follows:

$$\begin{aligned} & \text{Minimise} && (\mathbf{c}^P)^T \mathbf{x}^P + (\mathbf{c}^R)^T \mathbf{x}^R + (\mathbf{c}^D)^T \mathbf{d} \\ & \text{subject to} && A^P \mathbf{x}^P = \mathbb{1} \\ & && A^R \mathbf{x}^R = \mathbb{1} \\ & && B^P \mathbf{x}^P - B^R \mathbf{x}^R \leq 0 \\ & && D^P \mathbf{x}^P - D^R \mathbf{x}^R - \mathbf{d} \leq 0, \end{aligned} \quad (3)$$

where $\mathbf{x}^P \in \{0, 1\}^{n^P}$, $\mathbf{x}^R \in \{0, 1\}^{n^R}$, $\mathbf{d} \in \{0, 1\}^{m^D}$ are binary variables and $\mathbf{c}^D \in \mathbb{R}_+^{m^D}$ is some positive penalty parameter. Variable d_i equals 1 if restricted connection i is operated by a crew but no aircraft and 0 otherwise. The first two sets of constraints are identical to the original problem formulations. The third set of constraints enforces that short connections which are operated by some crew are also operated by some aircraft. The last set of constraints provokes additional cost in the objective function if a restricted connection is operated by a crew and no aircraft. In Section 5 we find solutions for the integrated model applied to schedules of dated periods of one week.

3. Literature

Airline scheduling problems have been addressed in an extensive number of publications (see Klabjan [18] for a detailed overview). We list selected literature addressing airline scheduling problems and describe attempts to integrate the aircraft routing and crew pairing problems. We conclude with recent formulations that incorporate robustness measures.

We are not aware of contributions that discuss the schedule design problem as a single problem. Since schedule design is addressed in combination with fleet assignment we start with the literature on fleet assignment. The fleet assignment model has been formulated by Hane et al. [17] as a multi-commodity flow problem. It is called *leg-based* because revenue effects between flight-legs are not modelled. To take such network effects into account, Barnhart et al. [2] described an enhanced model using demand forecasts for origin–destination pairs.

An integrated model for schedule design and fleet assignment was presented by Barnhart [22]. They used the origin–destination fleet assignment model and flights are chosen from an optional set of flights to maximise profit.

The aircraft routing problem has been addressed in a number of publications, for example, in Clarke et al. [4], Feo and Bard [12], Daskin and Panayotopoulos [7] and Gopalan and Talluri [15] and more recently in Grönkvist [16]. Sarac et al. [31] considered the aircraft routing problem on an operational level rather than a planning level. They added constraints to the set partitioning formulation to satisfy maintenance availability at the maintenance bases. Due to these constraints they had to alter the branching strategy, as described in Section 2.2. They used a combination of follow-on (see Section 2.1) and aircraft–flight pair (see Section 2.2) branching.

After fuel costs, crew salary is the second largest operational cost an airline has to account for. Therefore finding a minimal cost solution to the crew pairing problem is important. On the other hand it is a very complicated problem due to the large number of possible pairings, the complicated rule structure and the necessity to find integer solutions. For these reasons the crew pairing problem has received a lot of attention in the literature, see Barnhart et al. [3] for a detailed description of the crew pairing problem and a review of the literature addressing the problem. Also recently, Gopalakrishnan and Johnson [14] gave a comprehensive overview on state-of-the-art methods to solve the crew pairing problem.

The crew rostering problem can be viewed as a separate optimisation problem with no effect on the cost of the integrated solution. We refer to Ernst et al. [11] for an annotated bibliography of rostering problems.

Various combinations of airline scheduling problems have been integrated in the literature. Here we focus on the integration of crew pairing and aircraft routing problems.

Klabjan et al. [19] partially integrated aircraft routing, crew pairing and schedule design. They reversed the order of the crew pairing and aircraft routing problems. Plane count constraints are added to the crew pairing problem to ensure the existence of a feasible solution for the aircraft routing problem. Plane count constraints ensure that at most the number of available aircraft is used at any time. Their results are based on a *hub-and-spoke network*. In this network only large airports (hubs) are linked by direct flights and all smaller airports (spokes) are only connected to one hub. Many aircraft meet at the same hub at the same time ensuring the existence of many feasible connections. This property leads to a much larger number of feasible routings than in an *interconnected network*. In interconnected networks many airports are linked with multiple other airports by direct flights. To include schedule design in the model the departure time of each flight is allowed to vary in some time window. This is done by relaxing feasibility parameters in the crew pairing problem and hence generating a larger set of pairings. Each feasible pairing then has a departure time attached to each flight contained in the pairing. Klabjan et al. [19] solved the crew pairing problem via a linear program based branch-and-bound algorithm.

Another model to integrate aircraft routing and crew pairing was proposed by Cordau et al. [6] and also by Mercier et al. [24]. They used Benders decomposition and branch-and-price to solve the model. Employing the crew pairing problem as the subproblem as well as

the master problem has been tested, the latter with better success. Both approaches add inequalities to the set partitioning polytopes of the problems. Cordau et al. [6] also reversed the sequential approach and tried to solve the crew pairing problem first followed by the aircraft routing problem as in Klabjan et al. [19]. They applied this approach to an interconnected network but were not successful in obtaining feasible solutions for the aircraft routing problem.

Cohn and Barnhart [5] also integrated aircraft routing and crew pairing. They extended the crew pairing problem by using the aircraft routing problem as a second column generator next to the crew pairing generator. For each solution of the aircraft routing problem one variable is added to the crew pairing problem and a convexity constraint ensures the selection of one of the aircraft routing solutions in the final solution of the problem. LP based branch-and-price is used in this computationally expensive solution method. Mercier et al. [24] found that their Benders decomposition approach yields better solutions in less computation time than the extended crew pairing model of Cohn and Barnhart [5].

Sandhu and Klabjan [30] partially integrated fleet assignment, aircraft routing and crew pairing with a similar approach as Klabjan et al. [19] and solved the model with both Lagrangian relaxation and Benders decomposition.

Very recently, Mercier and Soumis [23] extended their model (Mercier et al. [24]) and integrated aircraft routing and crew pairing with time windows for the departure times. Flights are allowed to depart 5 min earlier or later than originally scheduled. Binary variables are used to indicate which departure time is assigned to a flight. Equality constraints sum up the binary departure time variables for the crew and aircraft solutions and ensure that the same departure times are used in the solutions of both problems. Again, the authors used Benders decomposition to solve the problem.

Models that focus purely on minimising cost tend to generate solutions that appear brittle in operations. Such solutions incur large recovery costs once disruptions occur in practice. In order to improve the behaviour in operations a number of robustness measures have been introduced.

Schaefer et al. [32] used expected operational cost for the crew pairings instead of planned cost. Interactive effects between pairings are ignored and a push-back strategy for recovery is used. In this strategy the flights are delayed until crew and aircraft are available. The authors used SimAir to estimate the costs and to evaluate the quality of their solutions. SimAir is a Monte Carlo simulation of airline operations that permits the evaluation of schedules and recovery policies in operations, see Rosenberger et al. [27].

Yen and Birge [36] formulated the crew pairing problem as a stochastic programming problem in a computationally expensive approach. Crew switching aircraft are penalised in the objective function. A similar measure of robustness was introduced by Ehrgott and Ryan [10] in a deterministic approach. Crew pairings are penalised where crew are changing aircraft and the sit-time of the crew is less than the minimal sit-time plus some measure of delay of the incoming flight. Crew who stay on the same aircraft are not penalised. Thus, crew connections where disruptions are likely to propagate onto multiple flights are penalised. Robustness is treated as a second objective function in a bicriteria approach. Mercier et al. [24] also penalised crew changing aircraft on restricted connections (see Section 2.3).

Most recently, Shebalov and Klabjan [33] solved the crew pairing problem first and then maximised the number of move-up crews without increasing the planned cost too much. Move-up crews are crews that can potentially be swapped in case one crew is delayed. They compared their method with the method of solving the standard crew pairing problem by simulating disruptions and found solutions with significantly lower operational costs if the additional cost allowed for move-up crews is not too high.

A direct comparison between the various approaches is difficult due to different levels of integration, robustness measures and characteristics of schedules and rule-sets used. Cordeau et al. [6] and Mercier et al. [24] found their Benders decomposition approach superior to two recent models (Klabjan et al. [19] and Cohn and Barnhart [5]). It is also evident that a direct solution approach to the integrated model (3) for large scale practical problems is inferior to decomposition techniques if not intractable (Cordeau et al. [6]).

4. Iterative solution approach

Currently, the most successful approach in the literature to solve the integrated crew pairing and aircraft routing problem seems to be Benders decomposition (see Mercier et al. [24]). However, large computation time is needed and the problem is only solved to near optimality.

We assume the aircraft routing cost to be fixed and consider the cost of the crew pairings as the only cost of the integrated solution. We consider a connection to be restricted, if the sit-time does not exceed the minimal sit-time by more than 30 min. The non-robustness measure (NRM) of an integrated solution is a sum of sit-time dependent penalties over all restricted aircraft changes. We choose penalties which increase linearly with decreasing sit-time:

$$NRM = \sum_{ij \in RAC} (k_1 - ((\text{sit-time})_{ij} - (\text{minimal sit-time})_{ij}))k_2, \quad (4)$$

where RAC is the set of all restricted aircraft changes. Weights k_1 and k_2 are chosen such that the penalty equals 7 for a restricted aircraft change with sit-time equal to the minimal sit-time, 6 for a restricted aircraft change with sit-time exceeding the minimal sit-time by 5 min, and so on until a weight of 1 is assigned to a restricted aircraft change with sit-time exceeding the minimal sit-time by 30 min. Note that all departure and arrival times in the schedules we consider are discretised in 5 min intervals. A different set of connections (e.g. those with a larger sit-time) or a different function of the sit-time (e.g. where weights increase exponentially with decreasing sit-time) could be chosen in a straightforward way.

We search for an integrated solution with small cost and small non-robustness measure. We do not attempt to solve the integrated problem to optimality. Instead we propose to solve the crew pairing problem and the aircraft routing problem iteratively. Initially we solve the crew pairing problem to cost optimality without considering any aircraft routings. Not considering any aircraft routings results in the generation of a larger set of feasible crew pairings since feasibility parameters are relaxed (see Section 2.3). This solution yields a lower bound on the cost of a feasible integrated solution. Then, in each iteration the aircraft routing problem is solved first. We consider all restricted connections operated in the current crew pairing solution and force the aircraft routing solution to contain as many of those connections as possible. This will enforce the “aircraft to follow the crew” as much as possible if the connection is restricted. In other words, we solve the aircraft routing problem using the following objective function:

$$\max \sum_{ij \in RC^P \cap RC^R} (k_1 - ((\text{sit-time})_{ij} - (\text{minimal sit-time})_{ij}))k_2, \quad (5)$$

where RC^P is the set of restricted connections operated in the current crew pairing problem and RC^R the set of restricted connections used in the aircraft routing solution and k_1, k_2 are chosen as above. Note that we maximise over the set of restricted connections. This is in contrast to the non-robustness measure (4) where we sum over the set of restricted aircraft changes. Next we solve the crew pairing problem to optimality for the current aircraft routing solution with

a weighted sum objective function of crew pairing costs and non-robustness measure:

$$\min \text{crewPairingCost} + \text{crewPenalty} \times \text{NRM}. \quad (6)$$

Hence, each iteration yields a feasible solution to the integrated problem. We start with *crewPenalty* equal to zero and increase this weight in each iteration in order to increase the robustness of the solutions we generate.

The advantages of this procedure are as follows: firstly, we can generate solutions with small cost and small non-robustness measure in a short amount of computation time and with no increase of complexity compared to the original problems. Secondly, we can study the trade-off between crew pairing costs and non-robustness measure of the solutions and the airline can choose a preferred solution without attaching a cost to the non-robustness measure a priori. Finally, we obtain a lower bound for the crew pairing cost which allows us to measure the quality of the final solution. Algorithm 1 shows the steps of the iterative approach.

Algorithm 1. Iterative algorithm

- 1: SET *crewPenalty* = 0
- 2: SOLVE crew pairing problem with objective function (6) (since no aircraft routings are taken into account a larger set of feasible pairings is generated).
- 3: **while** *crewPenalty* \leq *maxCrewPenalty* **do**
- 4: SOLVE aircraft routing problem with objective function (5) (maximise the number of restricted connections contained in the aircraft routing solution that are operated in the current crew pairing solution).
- 5: SOLVE crew pairing problem with objective function (6) (minimise cost and the number of restricted aircraft changes).
- 6: INCREASE *crewPenalty*
- 7: BREAK if the non-robustness measure cannot be improved
- 8: **end while**

For the schedule data sets relevant to this work, it can be assumed that the minimal sit-time of the crew is equal to the minimal turn-time of aircraft. Hence no short connections are taken into account. Since we always solve the crew pairing problem for a given aircraft routing solution, short connections can be considered by removing connections in the underlying network structure of the crew pairing problem, i.e. short connection arcs are removed that are not operated by any aircraft. In the aircraft routing problem short connections can be included by assigning very large weights to short connections operated by the crew in objective (5). If short connections are present in the problem, Step 2 of Algorithm 1 generally yields an infeasible solution that violates the short connection rules (Section 2.3). The interdependence between aircraft routings and crew pairings stated in Section 2.3 is extended by an additional rule. We understand that this rule has been imposed by the airline in an attempt to improve robustness. The number of times the crew are allowed to change aircraft in a single duty period is restricted to one for technical crew and two for flight attendants. In the following discussion we refer to these restrictions by DPACLIM (duty period aircraft change limit).

The cost of the crew pairing solution of Step 2 yields a lower bound on the crew pairing costs of a feasible integrated solution since no aircraft routings are taken into account. In our experiments, no aircraft routing solution for the crew pairing solution of Step 2 could be found to satisfy the DPACLIM rule. Hence, Step 2 usually yields an infeasible solution to the integrated problem. If the DPACLIM rule is relaxed we can find an aircraft routing solution to form a feasible solution to the integrated problem. For the test instances, such an integrated solution contains a large number of restricted aircraft changes and hence accounts for a large non-robustness measure.

The DPACLIM rule is implicitly enforced by the shortest path column generator of the crew pairing problem since we always solve the crew pairing problem for a given aircraft routing solution.

After the initial steps of the algorithm, we obtain a feasible solution to the integrated problem in each iteration by solving the crew pairing problem (Step 5) for a given aircraft routing solution (Step 4). Once the integrated solution converged to a stable solution, the algorithm stops. For a stable solution, successive iterations yield identical aircraft routing and crew pairing solutions, despite increasing the *crewPenalty* weight. Hence, the non-robustness measure cannot be improved. The value of *maxCrewPenalty* is chosen such that the non-robustness measure dominates the crew pairing costs in function (6) in the sense that no restricted aircraft changes are contained in the optimal solution if such a solution exists.

From a computational point of view, the main advantage of this solution algorithm to the integrated problem is that the original set partitioning formulations of both problems remain unchanged. This preserves the properties of the two problems and enables us to solve each problem efficiently with the methods described in Section 2.

If aircraft routing and crew pairing solvers are available, the implementation of the iterative approach is straightforward. We influence the characteristics of the solutions to either problem via the costs of the underlying network structure. This is easily implemented into the shortest path computations of the column generators for both problems: penalties are applied to restricted connections used by the current crew solution in the network for the aircraft routing calculations. Since there are no other costs associated with the aircraft routings these penalties will enforce paths in the solution that contain as many restricted crew connections as possible. Also, we always obtain a feasible aircraft routing solution since all changes to the problem are in the objective function. In the crew pairing problem, each restricted aircraft change will incur a penalty during the shortest path calculation. In Section 5.4 we apply the iterative algorithm to a number of different data sets.

The crew for an aircraft usually consists of multiple crew groups. Our data sets consist of three crew groups, namely captains and first officers (i.e. technical crew) and flight attendants. Different rules, base strengths and pay structures apply to each group. A robust aircraft routing and crew pairing solution for one crew group may enforce many restricted aircraft changes in a crew pairing solution for another crew group. Hence, considering aircraft and one crew group might lead to a suboptimal solution. Ideally we want to consider all crew groups and the aircraft simultaneously. To incorporate multiple crew groups into the iterative approach we simply solve the crew pairing problem in Step 5 for each crew group separately but use the same common aircraft routing solution. To obtain penalties for the restricted connections for the subsequent aircraft routing problem we weight the penalties for the restricted connections of the different crew pairing solutions according to weights (w_1 and w_2 in example (7)) chosen by the airline. For example, the aircraft routing objective function in Step 4 for two crew groups is changed to

$$\begin{aligned} \max w_1 \sum_{ij \in RC_1^p \cap RC^R} (k_1 - ((\text{sit-time})_{ij} - (\text{minimal sit-time})_{ij}))k_2 \\ + w_2 \sum_{ij \in RC_2^p \cap RC^R} (k_1 - ((\text{sit-time})_{ij} - (\text{minimal sit-time})_{ij}))k_2, \quad (7) \end{aligned}$$

where RC_1^p and RC_2^p are the sets of restricted connections of the crew pairing solutions for crew group 1 and crew group 2, respectively. Weights w_1 and w_2 can be chosen to reflect the ratio of crew pairing costs between both crew groups. We then generate a new aircraft routing solution as before. The results of the *iterative algorithm with multiple crew groups* are presented in Section 5.5.

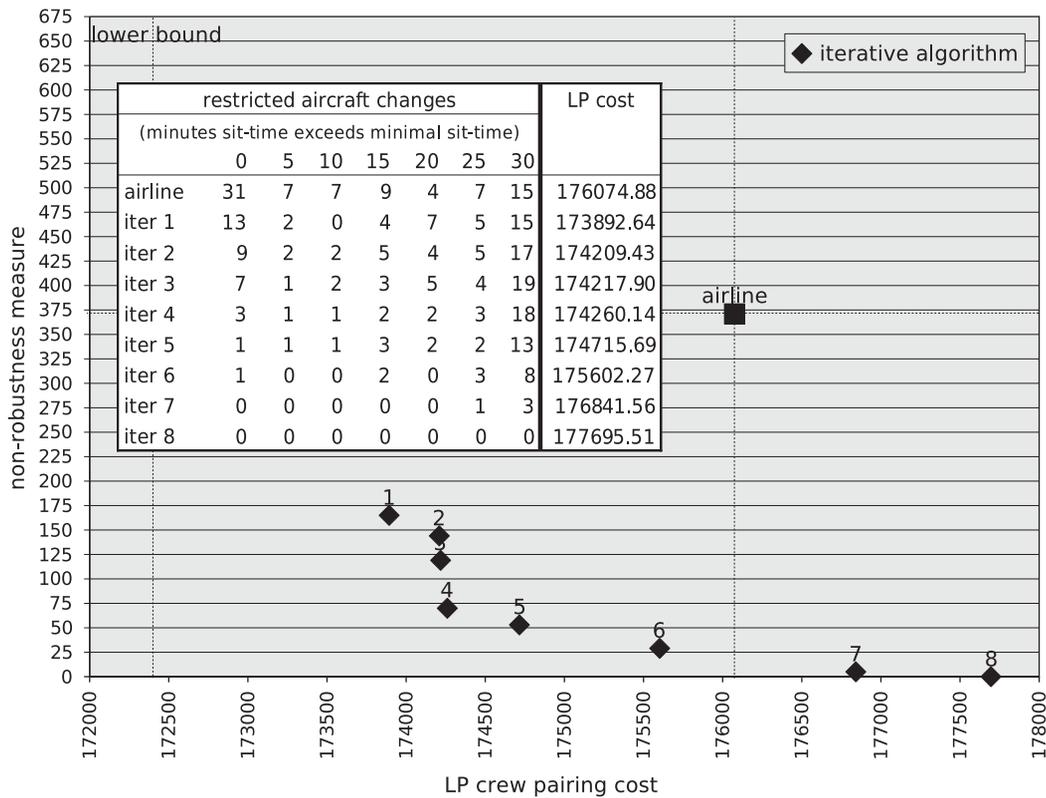


Fig. 1. Comparison of airline and iterative approach, captain solutions (DPA CLIM = 1) for schedule summer 2005.

5. Computational experiments

5.1. Flight schedule

The computational experiments are performed on interconnected flight networks corresponding to domestic airline schedules. The schedules vary on a daily basis and we consider dated time periods of one week. We consider three different schedules, from summer 2005, winter 2005 and summer 2006. For each schedule we compute solutions to the integrated aircraft routing and crew pairing problem for three different crew groups: captains, first officers and flight attendants. Each crew group has three different crew bases with different strengths at each base. Also, meal-break rules and rest requirements as well as the pay structure vary between the groups.

Each schedule contains approximately 750 flights and 22,000 connections in one week and the fleet consists of 14 aircraft. Approximately 1200 of the connections are restricted in the sense that the sit-time exceeds the minimal sit-time by at most 30 min. All rules and restrictions applicable to aircraft routings and crew pairings are provided by the airline and are implemented in the algorithm. Hence, all solutions generated satisfy these rules and are ready to be used in practice.

Considering dated schedules of one week is in line with current practice at the airline because resource levels at the crew bases as well as the schedule itself change on a weekly basis. In order to compare the iterative approach solutions with solutions currently operated we restrict the computational experiments to dated weekly schedules. Other problems such as the cyclic daily or weekly problem (where pairings and routings wrap around the same day or week) or dated monthly schedules (Klabjan [18]) can be considered by modifying the crew pairing and aircraft routing algorithms, the general method of the iterative approach remains the same.

5.2. Crew pairing

As a measure of the cost of the integrated solution we use crew pairing costs without added penalties (see objective function (6)). In all figures, LP crew pairing costs rather than IP costs are presented to obtain a more monotonic representation of the solutions. In all calculations the observed IP-gap is very small, with an average of 0.81% over all solutions generated.

Because many of the crew pairing columns generated in each solution of the iterative approach remain legal and feasible for subsequent crew pairing solutions, we can store the columns and simply recalculate their objective coefficients in each iteration. Each LP computation is “hot-started” from the optimal basis of the LP solution of the previous iteration. As a result the crew pairing solutions are obtained very efficiently during the later iterations of the algorithm since very few new columns are generated.

5.3. Aircraft routing

In the schedules considered, aircraft maintenance is usually performed at night when all aircraft are grounded at few airports. If an aircraft stays overnight at an airport with no maintenance facility, this aircraft must overnight at a maintenance base the next night. Also, there is no interaction between connections performed by the aircraft on different days. Hence, we can solve the aircraft routing problem for each day independently taking the maintenance status of the preceding night into account.

5.4. Iterative algorithm with one crew group

In Fig. 1 results of the captain solutions for the schedule of summer 2005 are presented. The costs and non-robustness measures of

the solution operated by the airline and the solutions generated by the iterative approach are compared. The airline solution is generated by solving the crew pairing problem to cost optimality for the aircraft routing that was operated by the airline. The figure shows the non-robustness measure (see Section 4) plotted against the optimal LP crew pairing cost. In addition we list the number of restricted aircraft changes for each solution. The solutions generated by the iterative approach yield better costs and non-robustness measures than the airline solution. These results seem surprising since we expect to generate solutions that are more robust but not cheaper than the airline solution. The same crew pairing optimiser is used for airline solution and iterative approach solutions and the only differences between them are the aircraft routings used as input. The airline solution uses an aircraft routing solution that is constructed manually with no relation to the crew pairing problem. In the iterative approach the aircraft routing solutions are fitted closely to the cheapest possible crew pairing solution (from Step 2 of Algorithm 1). These aircraft routing solutions permit the construction of crew pairing solutions that are cheaper and more robust than the solution that is currently used by the airline because many characteristics of cheap crew pairing solutions are incorporated in these aircraft routing solutions.

The data shown in Fig. 1 is given in more detail in Table 1. For each iteration we list the LP crew pairing cost and the improvement compared to the LP crew pairing cost of the airline solution in percent. We also list the IP crew pairing costs and the IP-gaps. Note that the crew pairing LP uses a weighted sum objective of crew pairing costs and penalties (see objective function (6)). The ratio of costs and penalties in the solution value of LP and corresponding IP solution can differ. When we subtract the penalties from the solution values we may obtain an IP solution that incurs less crew pairing cost than the LP solution. Additionally, the non-robustness measures and the improvements relative to the airline solution are shown. We do not show any non-robustness measure for the lower bound solution since we cannot find an integrated solution satisfying the DPACLIM rule. Finally, CPU times are shown for each iteration. All experiments are performed on a Pentium 4, 2.6GHz computer. For all solutions only the time for solving the crew pairing problem is shown since the aircraft routing problem is always solved in less than 5 s due to the applied decomposition described in Section 5.3. We can observe that the time needed for the later iterations is significantly smaller than for the early iterations. This speed-up is caused by using previously generated crew pairings and hot-starting from a previous basis solution.

The cost of the crew pairing solution obtained in Step 2 of Algorithm 1 is a lower bound for the cost of the optimal solution value. In Table 1 we can observe for the captain solutions, which this lower bound solution incurs 2.08% less cost than the airline solution. The solution is infeasible since we cannot find aircraft routings to satisfy the DPACLIM rule. Here, only one aircraft change is allowed during each duty period. The cheapest feasible solution we find (iteration 1) incurs 1.24% less cost than the airline solution and is only 0.85% more expensive than the value of the lower bound. Also, the non-robustness measure of this solution is only 165 compared to 371 for the airline solution which is an improvement of 55.53%. The most robust solution that is still cheaper than the airline solution improves the non-robustness measure by 92.18% (iteration 6). For first officers results look very similar (see Table 1). Here the non-robustness measure can be improved by 97.09% (iteration 7) without increasing the cost of the airline solution.

In Fig. 2 we demonstrate the benefit of relaxing the DPACLIM rule allowing at most two aircraft changes per duty period instead of just one. We find solutions with costs effectively on the lower bound (costs increase by 0.05% for iterations 1 and 2) that are also very robust. In Table 2 we list details for both DPACLIM settings.

We list the number of duty periods containing one (DPAC1) and two (DPAC2) aircraft changes, respectively. We also list the number of aircraft changes occurring in these duty periods and how many of those are restricted aircraft changes. We complement these figures with the corresponding LP crew pairing costs and non-robustness measures. We observe that the reduction in the cost objective is achieved with an increased number of duty periods with two aircraft changes. However, the non-robustness measure of these solutions does not increase significantly. In iteration 6, for example, the solution with one allowed aircraft change has a non-robustness measure of 29 and 14 restricted aircraft changes, while the solution with two allowed aircraft changes per duty period has a non-robustness measure of 32 and also 14 restricted aircraft changes. The latter solution incurs 0.76% less costs but contains nine duty periods with two aircraft changes. This shows that we can achieve a better cost and more robust

In Tables 3–5 we present further results for technical crew and flight attendants for schedules of summer 2005 and 2006, and winter 2005, respectively. The overall trend is similar to the captain solutions for summer 2005 shown in Table 1. We find solutions that are much more robust than the airline solution and incur less or only slightly more cost. We find that for the flight attendant solutions for summer 2006 the non-robustness measure does not decrease below 50 (see Table 5). This is due to restrictive base constraints. In experiments where the base constraints are relaxed and the number of available crew is increased by 2%, solutions with a non-robustness measure of zero can be obtained. Note also that for the schedules of summer 2006 there exists only one set of results for technical crew since the rule-sets for captains and first officers are identical for this period. Additionally, the costs of the iterative solutions and of the airline solution are much closer to the lower bound for flight attendants than for technical crew. This is because of the relaxed DPACLIM rule for flight attendants where two aircraft changes in each duty period are allowed by the airline. Finally, we observe in Table 3 large computation times in iterations 3 and 7 for the first officer scenario. In both cases the branch-and-bound search in the crew pairing problem is stopped after 1000 nodes. The integer solutions shown are found in an early stage of the search but their solution values exceed the bound gap.

It is desirable to compare the performance of the iterative solution approach with a direct solution method for the integrated model (3) as well as other decomposition approaches. We omit the direct solution method since it is commonly believed in the literature that decomposition methods outperform a direct solution approach for large scale practical problems (Cordau et al. [6]). A comparison with other decomposition approaches is complicated by different types of schedules and rule-sets used in the various publications and not included in the paper.

5.5. Iterative algorithm with multiple crew groups

We extend the iterative algorithm to solve the aircraft routing problem together with the crew pairing problems for technical crew and flight attendants. The goal is to show that the benefits of solving one crew group together with the aircraft are not lost if we add another crew group. Clearly, using a solution that is very robust for one crew group but results in many restricted aircraft changes for another crew group is not desirable. Figs. 3 and 4 summarise the results. We choose the schedule of summer 2006 since this schedule only contains two different crew groups, technical crew and flight attendants. In Fig. 3 the results of applying the iterative algorithm to technical crew only are shown in grey. Similarly, in Fig. 4 the results of considering flight attendants only are also marked in grey. Additionally, we can solve the technical crew as before and in each

Table 1
Results for technical crew, schedule summer 2005.

	Captain, schedule summer 2005							First officer, schedule summer 2005						
	LP cost	Imp. (%)	IP cost	Gap (%)	NRM	Imp. (%)	CPU time*	LP cost	Imp. (%)	IP cost	Gap (%)	NRM	Imp. (%)	CPU time*
Airline	176,074.88	–	177,647.70	0.89	371	–	605	172,510.58	–	173,176.49	0.38	344	–	408
Bound	172,420.84	2.08	173,854.41	0.82	–	–	724	168,419.26	2.37	170,560.04	1.26	–	–	538
Iter. 1	173,892.64	1.24	175,722.08	1.04	165	55.53	420	170,079.29	1.41	171,978.93	1.10	208	39.53	413
Iter. 2	174,209.43	1.06	175,626.14	0.81	144	61.19	170	170,274.45	1.30	171,831.82	0.91	165	52.03	387
Iter. 3	174,217.90	1.05	175,187.69	0.55	119	67.92	360	170,544.68	1.14	171,232.97	0.40	113	67.15	257
Iter. 4	174,260.14	1.03	174,984.87	0.41	70	81.13	227	170,260.76	1.30	170,695.24	0.25	73	78.78	223
Iter. 5	174,715.69	0.77	175,707.72	0.56	53	85.71	275	170,388.66	1.23	170,908.26	0.30	30	91.28	233
Iter. 6	175,602.27	0.27	176,311.24	0.40	29	92.18	306	170,941.18	0.91	171,527.42	0.34	17	95.06	193
Iter. 7	176,841.56	–0.44	177,265.20	0.24	5	98.65	188	172,009.62	0.29	172,418.49	0.24	10	97.09	222
Iter. 8	177,695.51	–0.92	178,066.47	0.21	0	100.00	256	173,395.12	–0.51	173,329.55	–0.04	2	99.42	245
Iter. 9								173,909.61	–0.81	174,183.32	0.16	0	100.00	194

*For solving the crew pairing problem; all times in seconds.

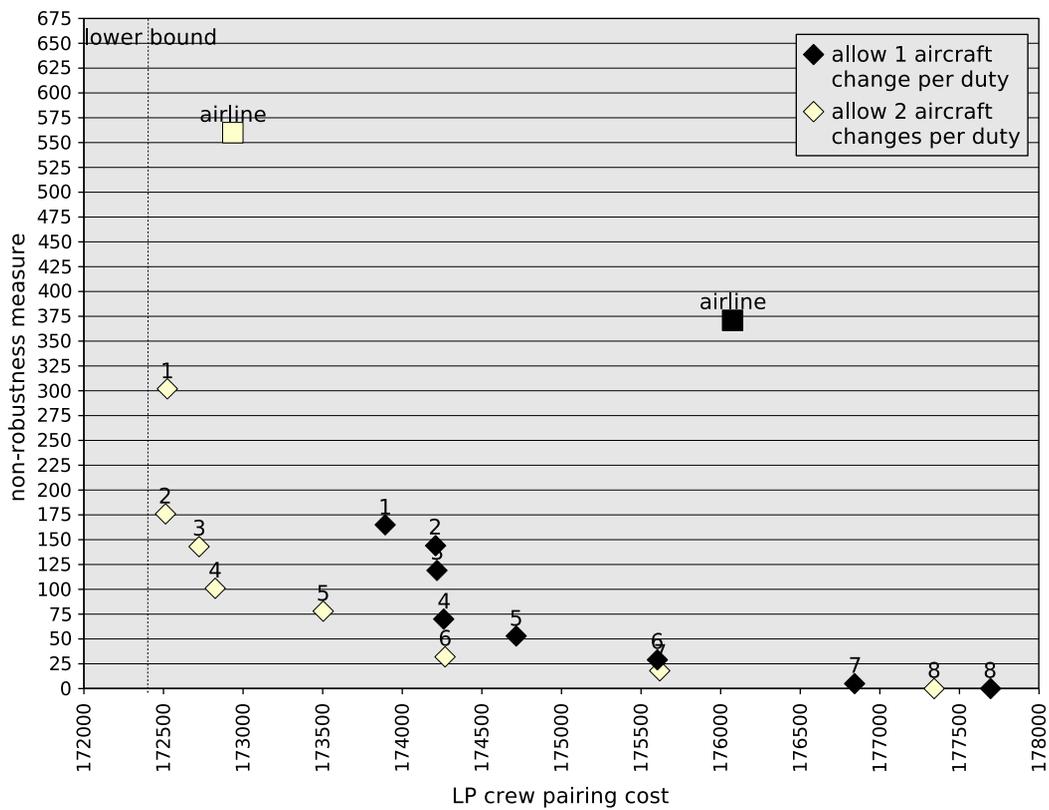


Fig. 2. Variation of DPAC settings (black = 1; white = 2), captain solutions for schedule summer 2005.

Table 2
Impact of allowed aircraft changes per duty period (DPAC) for captain solutions, schedule summer 2005.

	Allow one aircraft change					Allow two aircraft changes							
	DPAC1*	AC*	RAC*	LP cost	NRM*	DPAC1	AC	RAC	DPAC2*	AC	RAC	LP cost	NRM
Airline	103	103	66	176,074.88	371	76	76	55	52	104	67	172,935.52	560
Iter. 1	85	85	34	173,892.64	165	73	73	36	25	50	27	172,524.21	302
Iter. 2	80	80	36	174,209.43	144	59	59	20	21	42	19	172,512.80	176
Iter. 3	83	83	32	174,217.90	119	69	69	24	16	32	12	172,723.57	143
Iter. 4	76	76	22	174,260.14	70	58	58	18	14	28	12	172,825.09	101
Iter. 5	63	63	17	174,715.69	53	55	55	15	8	16	8	173,503.36	78
Iter. 6	70	70	14	175,602.27	29	66	66	8	9	18	6	174,268.66	32
Iter. 7	71	71	3	176,841.56	5	55	55	3	14	28	4	175,617.27	18
Iter. 8	71	71	0	177,695.51	0	60	60	0	15	30	0	177,341.25	0

*DPAC1—duty periods with one aircraft change; DPAC2—duty periods with two aircraft changes; AC—aircraft changes; RAC—restricted aircraft changes; NRM—non-robustness measure.

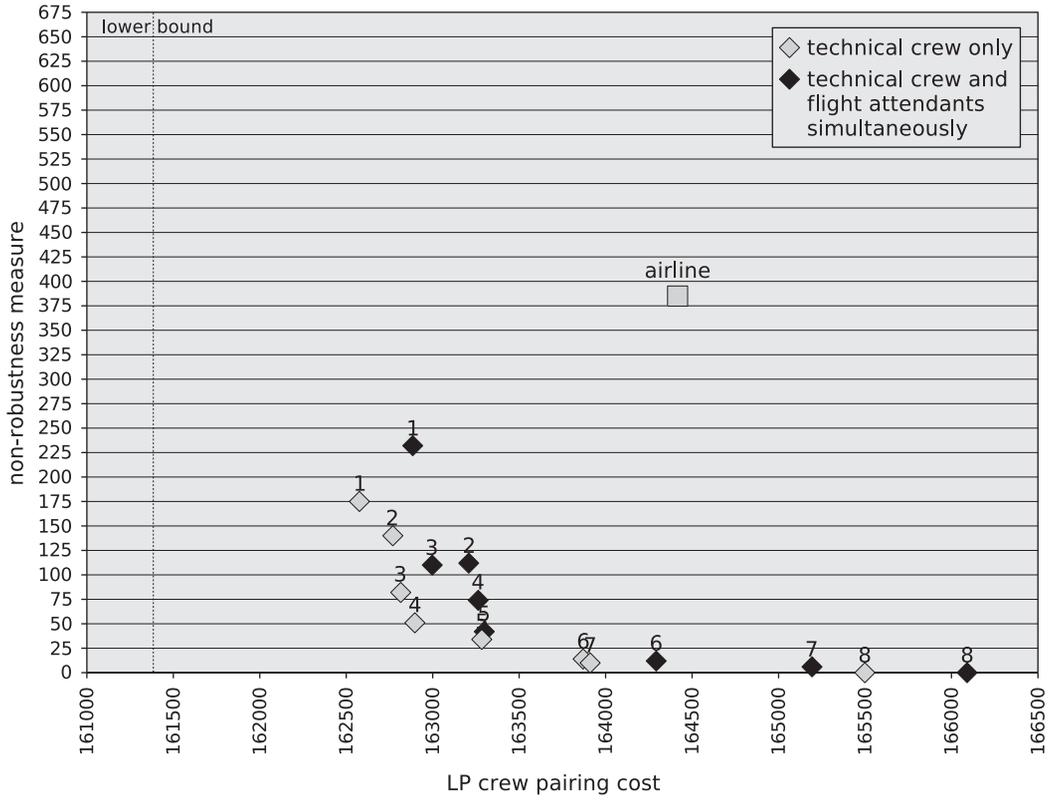


Fig. 3. Results for technical crew, schedule summer 2006, with flight attendants solved simultaneously.

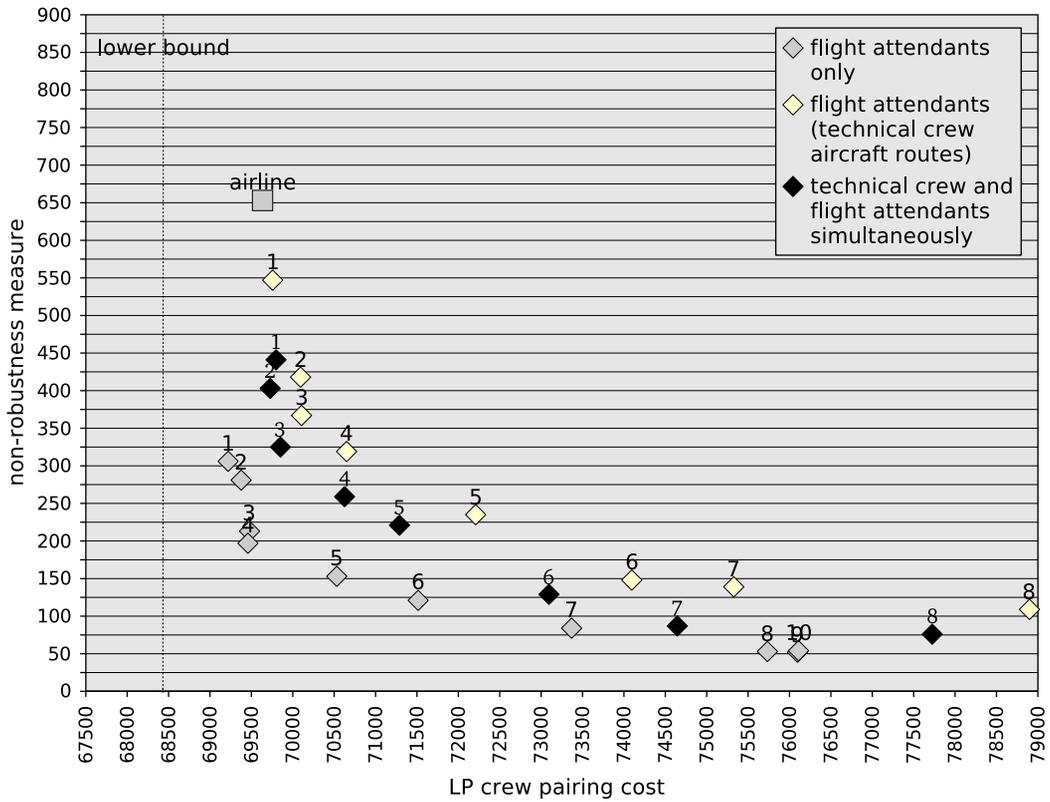


Fig. 4. Results for flight attendants, schedule summer 2006, with technical crew solved simultaneously.

Table 3
Results for technical crew, schedule winter 2005.

	Captain, schedule winter 2005							First officer, schedule summer 2005						
	LP cost	Imp. (%)	IP cost	Gap (%)	NRM	Imp. (%)	CPU time*	LP cost	Imp. (%)	IP cost	Gap (%)	NRM	Imp. (%)	CPU time*
Airline	170,767.52	–	171,428.20	0.39	443	–	632	166,361.00	–	168,440.67	1.23	374	–	420
Bound	168,153.35	1.53	168,765.37	0.36	–	–	805	162,590.00	2.27	163,235.40	0.40	–	–	612
Iter. 1	169,584.85	0.69	170,965.03	0.81	267	39.73	507	165,005.84	0.81	165,997.41	0.60	199	46.79	335
Iter. 2	169,542.08	0.72	170,473.83	0.55	148	66.59	350	164,591.74	1.06	164,824.08	0.14	146	60.96	227
Iter. 3	169,696.28	0.63	169,922.62	0.13	121	72.69	255	164,677.32	1.01	168,909.76	2.51	97	74.06	3996
Iter. 4	169,838.21	0.54	170,471.04	0.37	91	79.46	264	165,116.48	0.75	166,160.70	0.63	86	77.01	292
Iter. 5	169,956.76	0.47	170,514.41	0.33	76	82.84	343	165,500.89	0.52	167,020.75	0.91	65	82.62	350
Iter. 6	171,045.32	–0.16	171,700.82	0.38	24	94.58	349	166,284.73	0.05	166,351.09	0.04	41	89.04	255
Iter. 7	172,390.64	–0.95	172,879.06	0.28	19	95.71	394	166,966.68	–0.36	167,824.08	0.51	23	93.85	2386
Iter. 8	172,787.03	–1.18	174,345.83	0.89	1	99.77	440	168,875.44	–1.51	170,254.06	0.81	4	98.93	397
Iter. 9	173,217.90	–1.43	173,502.53	0.16	1	99.77	227	169,788.78	–2.06	170,915.92	0.66	2	99.47	351
Iter. 10	173,201.64	–1.43	173,813.00	0.35	1	99.77	296	170,249.68	–2.34	170,585.82	0.20	1	99.73	216

*For solving the crew pairing problem; all times in seconds.

Table 4
Results for flight attendants, schedules summer and winter 2005.

	Flight attendants, schedule summer 2005							Flight attendants, schedule winter 2005						
	LP cost	Imp. (%)	IP cost	Gap (%)	NRM	Imp. (%)	CPU time*	LP cost	Imp. (%)	IP cost	Gap (%)	NRM	Imp. (%)	CPU time*
Airline	68,787.11	–	69,833.49	1.50	497	–	146	68,425.19	–	70,046.91	2.32	511	–	484
Bound	68,671.77	0.17	69,547.70	1.26	–	–	138	67,909.60	0.75	70,780.45	4.06	–	–	1945
Iter. 1	68,749.99	0.05	70,331.80	2.25	256	48.49	109	68,142.56	0.41	70,175.15	2.90	181	64.58	2514
Iter. 2	68,834.40	–0.07	70,285.10	2.06	129	74.04	89	68,214.09	0.31	69,025.95	1.18	149	70.84	84
Iter. 3	68,969.46	–0.27	69,689.90	1.03	115	76.86	83	68,302.00	0.18	69,529.05	1.76	127	75.15	70
Iter. 4	69,161.13	–0.54	70,263.65	1.57	75	84.91	82	68,555.44	–0.19	69,623.45	1.53	95	81.41	112
Iter. 5	69,473.98	–1.00	69,750.55	0.40	45	90.95	207	68,778.60	–0.52	69,837.30	1.52	92	82.00	92
Iter. 6	70,230.34	–2.10	71,525.90	1.81	35	92.96	104	70,683.56	–3.30	71,186.90	0.71	62	87.87	72
Iter. 7	71,926.64	–4.56	72,857.50	1.28	9	98.19	88	71,549.44	–4.57	73,300.70	2.39	28	94.52	606
Iter. 8	73,313.53	–6.58	73,988.75	0.91	1	99.80	84	74,702.32	–9.17	75,125.70	0.56	10	98.04	154
Iter. 9	73,289.27	–6.55	74,495.55	1.62	0	100.00	75	74,827.86	–9.36	75,983.30	1.52	6	98.83	82
Iter. 10								75,073.00	–9.72	75,321.95	0.33	5	99.02	615

*For solving the crew pairing problem; all times in seconds.

Table 5
Results for technical crew and flight attendants, schedule summer 2006.

	Technical crew, schedule summer 2006							Flight attendants, schedule summer 2006						
	LP cost	Imp. (%)	IP cost	Gap (%)	NRM	Imp. (%)	CPU time*	LP cost	Imp. (%)	IP cost	Gap (%)	NRM	Imp. (%)	CPU time*
Airline	164,416.19	–	167,319.42	1.74	385	–	378	69,634.79	–	70,905.51	1.79	653	–	143
Bound	161,398.87	1.84	161,514.29	0.07	–	–	413	68,440.59	1.71	68,795.93	0.52	–	–	140
Iter. 1	162,576.29	1.12	163,529.97	0.58	175	54.55	240	69,219.33	0.60	70,236.23	1.45	306	53.14	83
Iter. 2	162,769.47	1.00	165,425.37	1.61	140	63.64	291	69,376.38	0.37	70,044.08	0.95	281	56.97	65
Iter. 3	162,815.10	0.97	164,138.50	0.81	82	78.70	282	69,477.70	0.23	70,234.78	1.08	213	67.38	58
Iter. 4	162,896.94	0.92	163,631.86	0.45	51	86.75	175	69,458.04	0.25	69,919.93	0.66	197	69.83	53
Iter. 5	163,283.65	0.69	164,386.92	6.57	34	91.17	212	70,531.49	–1.29	70,393.98	–0.20	153	76.57	58
Iter. 6	163,869.72	0.33	164,330.03	0.28	14	96.36	139	71,514.31	–2.70	71,535.83	0.03	121	81.47	71
Iter. 7	163,910.29	0.31	164,187.37	0.17	10	97.40	172	73,368.54	–5.36	73,085.18	–0.39	84	87.14	73
Iter. 8	165,499.07	–0.66	166,085.32	0.35	0	100.00	225	75,733.24	–8.76	76,139.23	0.53	53	91.88	64
Iter. 9								76,097.23	–9.28	76,097.23	0.00	52	92.04	45

*For solving the crew pairing problem; all times in seconds.

iteration uses the aircraft routing solution to generate solutions for the flight attendant problem. The results are shown in white in Fig. 4. This corresponds to setting $w_1 = 1$ and $w_2 = 0$ in objective function (7). The solutions are more expensive and less robust than the solutions obtained from focusing on flight attendants and aircraft only. But the solutions follow the same pattern and are of good quality compared to the airline solution. Hence we do not need to sacrifice solution quality of the flight attendant problem in order to improve the technical crew solution and hence improve the overall solution to the integrated problem.

In a second experiment we again solve two crew pairing problems in each iteration but we now use feedback information from

the technical crew problem as well as the flight attendant problem to generate the aircraft routing solution. In the aircraft routing problem we weight the penalties for the restricted connections of the technical crew solution with a factor of 2 and the penalties for the restricted connections of the flight attendant solution with a factor of 1. This corresponds to setting $w_1 = 2$ and $w_2 = 1$ in objective function (7). We then generate aircraft routings subject to this objective function. The according solutions are represented in black in both figures. We observe that the technical crew solutions deteriorate slightly from the best solutions (grey). The solutions for the flight attendants are also not as good as the best flight attendant solutions (grey) but better than the solutions generated by not using any

feedback from the flight attendant solutions (white). Note that in these solutions both crew groups rarely change aircraft if the connection is restricted. This also implies that both crew groups usually stay together on restricted connections. This is of great advantage not only from a robustness point of view but also from an operational point of view. The solutions satisfy this property without the need to focus on the property in the algorithm. Since the technical crew solution usually incurs more cost than the flight attendant solution we only consider examples where the weight for the technical crew exceeds the weight for the flight attendants. The weights can be adjusted according to the airline's preferences. We can now generate solutions for the aircraft routing and two crew pairing problems in one integrated procedure. Likewise, additional crew groups or multiple aircraft types can be added.

6. Conclusion

Solving the integrated aircraft routing and crew pairing problem to optimality is computationally expensive. Also, to identify a robust solution, costs must be associated with non-robustness. In this paper we show that by heuristically coupling the two problems we can quickly generate a series of good quality solutions and do not need to attach costs to the non-robustness measure. Although optimality of the solutions cannot be guaranteed, we obtain solutions that incur less costs and are significantly more robust than solutions currently used in practice. We also find that the airline can operate cheaper and more robust solutions if two aircraft changes per duty period are allowed for technical crew. Additionally, we are able to consider multiple crew groups with only a minor modification of the algorithm.

Future research includes a comparison of the iterative approach with other approaches in the literature and the substitution of the crew pairing formulation with weighted sum objective by a superior bicriteria crew pairing approach (Ehrgott and Ryan [10]). We also plan to investigate solving the integrated problem to optimality and allow departure time windows for the flights to include schedule design decisions into the formulation. Departure times of the flights are allowed to vary in some window to create more flexibility for the generation of crew pairings and aircraft routings. This is expected to further improve cost and robustness of the integrated solution. To ensure that the same departure times are used by crew and aircraft and the set partitioning formulations are not disturbed we branch on the time windows which can be incorporated into the network structures of both problems.

References

- [1] Barnhart C, Johnson EL, Nemhauser GL, Savelsbergh MWP, Vance PH. Branch-and-price: column generation for solving huge integer programs. *Operations Research* 1998;46:316–29.
- [2] Barnhart C, Kniker T, Lohatepanont M. Itinerary-based airline fleet assignment. *Transportation Science* 2002;36:199–217.
- [3] Barnhart C, Cohn AM, Johnson EL, Klabjan D, Nemhauser GL, Vance PH. Airline crew scheduling. In: Hall RW, editor. *Handbook of transportation science*. Dordrecht: Kluwer Scientific Publishers; 2003.
- [4] Clarke LW, Johnson EL, Nemhauser GL, Zhu Z. The aircraft rotation problem. *Annals of Operations Research* 1997;69:33–46.
- [5] Cohn AM, Barnhart C. Improving crew scheduling by incorporating key maintenance routing decisions. *Operations Research* 2003;51:387–96.
- [6] Cordeau J-F, Stojković G, Soumis F, Desrosiers J. Benders decomposition for simultaneous aircraft routing and crew scheduling. *Transportation Science* 2001;35:375–88.
- [7] Daskin MS, Panayotopoulos ND. A Lagrangian relaxation approach to assigning aircraft to routes in hub and spoke networks. *Transportation Science* 1989;23: 91–9.
- [8] Day PR, Ryan DM. Flight attendant rostering for short-haul airline operations. *Operations Research* 1997;45:649–61.
- [9] Desaulniers G, Desrosiers J, Dumas Y, Marc S, Rioux B, Solomon MM, Soumis F. Crew pairing at Air France. *European Journal of Operational Research* 1997;97:245–59.
- [10] Ehrgott M, Ryan DM. Constructing robust crew schedules with bicriteria optimization. *Journal of Multi-Criteria Decision Analysis* 2002;11:139–50.
- [11] Ernst AT, Jiang H, Krishnamoorthy M, Owens B, Sier D. An annotated bibliography of personnel scheduling and rostering. *Annals of Operations Research* 2004;127:21–144.
- [12] Feo TA, Bard JF. Flight scheduling and maintenance base planning. *Management Science* 1989;35:1415–32.
- [13] Garey MR, Johnson DS. *Computers and intractability*. New York: Freeman; 1979.
- [14] Gopalakrishnan B, Johnson EL. Airline crew scheduling: state-of-the-art. *Annals of Operations Research* 2005;140:305–37.
- [15] Gopalan R, Talluri KT. The aircraft maintenance routing problem. *Operations Research* 1998;46:260–71.
- [16] Grönkvist M. Accelerating column generation for aircraft scheduling using constraint propagation. *Computers & Operations Research* 2006;33:2918–34.
- [17] Hane CA, Barnhart C, Johnson EL, Marsten RE, Nemhauser GL, Sigismundi G. The fleet assignment problem: solving a large-scale integer program. *Mathematical Programming* 1995;70:211–32.
- [18] Klabjan D. Large-scale models in the airline industry. In: Desaulniers G, Desrosiers J, Solomon MM, editors. *Column generation*. Dordrecht: Kluwer Scientific Publishers; 2005. p. 163–96.
- [19] Klabjan D, Johnson EL, Nemhauser GL, Gelman E, Ramaswamy S. Airline crew scheduling with time windows and plane count constraints. *Transportation Science* 2002;36:337–48.
- [20] Kohl N, Karisch S. Airline crew rostering: problem types, modeling and optimization. *Annals of Operations Research* 2004;127:223–57.
- [21] Lavoie S, Minoux M, Odier E. A new approach for crew pairing problems by column generation with an application to air transportation. *European Journal of Operational Research* 1988;35:45–58.
- [22] Lohatepanont M, Barnhart C. Airline schedule planning: integrated models and algorithms for schedule design and fleet assignment. *Transportation Science* 2004;38:19–32.
- [23] Mercier A, Soumis F. An integrated aircraft routing, crew scheduling and flight retiming model. *Computers & Operations Research* 2007;34:2251–65.
- [24] Mercier A, Cordeau J-F, Soumis F. A computational study of benders decomposition for the integrated aircraft routing and crew scheduling problem. *Computers & Operations Research* 2005;32:1451–76.
- [25] Minoux M. Column generation techniques in combinatorial optimization: a new application to crew pairing problems. In: *Proceedings of XXIVth AGIFORS symposium*, 1984.
- [26] Padberg MW. Perfect zero-one matrices. *Mathematical Programming* 1974;6:180–96.
- [27] Rosenberger JM, Schaefer AJ, Goldsman D, Johnson EL, Kleywegt AJ, Nemhauser GL. A stochastic model of airline operations. *Transportation Science* 2002;36: 357–77.
- [28] Ryan DM, Falkner JC. On the integer properties of scheduling set partitioning problems. *European Journal of Operations Research*, 1988.
- [29] Ryan DM, Foster BA. An integer programming approach to scheduling. In: Wren A, editor. *Computer scheduling of public transport urban passenger vehicle and crew scheduling*. Amsterdam: North-Holland; 1981. p. 269–80.
- [30] Sandhu R, Klabjan D. Integrated airline fleet and crew pairing decisions. *Operations Research*, 2006, to appear.
- [31] Sarac A, Batta R, Rump CM. A branch-and-price approach for operational aircraft maintenance routing. *European Journal of Operational Research* 2006;175(3):1850–69.
- [32] Schaefer AJ, Johnson EL, Kleywegt AJ, Nemhauser GL. Airline crew scheduling under uncertainty. *Transportation Science* 2005;39:340–8.
- [33] Shebalov S, Klabjan D. Robust airline crew pairing: move-up crews. *Transportation Science* 2006;40:300–12.
- [34] Vance PH, Barnhart C, Johnson EL, Nemhauser GL. Airline crew scheduling: a new formulation and decomposition algorithm. *Operations Research* 1995;45:188–200.
- [35] Wolsey L. *Integer programming*. New York: Wiley-Interscience; 1998.
- [36] Yen J, Birge J. A stochastic programming approach to the airline crew scheduling problem. *Transportation Science* 2006;40:3–14.