Modeling and Solving the Airline Schedule Generation Problem

A. Erdmann, A. Nolte, A. Noltemeier, R. Schrader

ZAIK, University of Cologne, Germany

October 7, 1999

Abstract

Since opening a new flight connection or closing an existing flight has a great impact on the revenues of an airline, the generation of the flight schedule is one of the fundamental problems in airline planning processes.

In this paper we concentrate on a special case of the problem. In contrast to airlines operating on regular schedules, the market for charter airlines is well-known and the schedule is allowed to change completely from period to period. Thus, precise adjustments to the demands of the market have a great potential for minimizing operating costs.

We propose a combined Branch-and-Cut approach to solve the airline schedule generation problem. To tighten the linear relaxation bound, we add cutting planes which adjust the number of aircraft and the spill of passengers to the demand on each itinerary.

For real-world problems from a large European charter airline we obtain solutions within a very few percent of optimality with running times in the order of minutes on a customary personal computer for most of the data sets.

Key Words: Capacitated network design; Dantzig-Wolfe decomposition; set partitioning, set packing; Branch-and-Cut

1 Introduction

Planning aircraft and crews is a very complex task. Thus, it is a current practice at airlines' planning departments to break up the resource planning into a number of stages which are then solved in a sequential manner. In the first stage, information on the market are collected and analyzed. The result of the *market analysis* is a demand estimation which is based on origin-destination pairs. The timetable is constructed in the next step. In this *schedule generation* step, the airline decides which direct flights should be offered in the new schedule. Also, the optimization of departure times is done, and operational constraints like fleet sizes are considered, too. Next, the aircraft type to use on each operated flight will be determined (*fleet assignment*). Determing aircraft routes (rotations) is the objective of the *tail assignment*. Using these aircraft rotations, pairings (duties) for yet unpersonalized crew members (cockpit and cabin crew) are determined in the *crew pairing*. Personalized monthly schedules are then constructed using the crew pairings in the *crew rostering*. Finally, very close to the day-of-operation some changes to the flight schedule may become necessary (*day-of-operation changes*).

Since the schedule generation problem (SGP) must be solved early in the airlines' overall planning process, its solution quality has a great influence on all subsequent steps. Most airlines, however, seem to construct their schedules manually. As this task is very complex and time consuming, it is common practice to only locally change the last schedule to adopt it to the new requirements. This also implies that the new schedule may inherit properties that are unnecessary and costly.

In this paper we address a special case of the schedule generation problem for the airline industry which arises particularly in charter business. In contrast to regularly operating carriers, charter airlines have more freedom to change their schedule from period to period. Furthermore, as in our case, there often are close business links to one or more tour operators who sell flights or even complete vacation packages (flight, hotel, rental car, etc.) to their customers. Then large contingents of the seats are booked before the schedule generation process is terminated, which, in addition to the data about customer behavior in the past, gives a very accurate knowledge about the market.

Our solution approach incorporates the most important operational constraints in the planning phase, like block times, minimum ground times, and curfews. Thus, our model combines schedule generation and fleet (or even tail) assignment.

2 Related Work

Network design problems have been intensively studied. Especially for problems arising in the telecommunications industry and in the context of freight transportation, numerous successful algorithms have been developed. Solution strategies include techniques based on linear programming, Lagrangean relaxation and Bender's decomposition, commonly used in combination with branch-and-bound (cf., e.g. Magnanti et al. [17], [18], Holmberg and Hellstrand [13], Holmberg and Yuan [14], Balakrishnan et al. [2], Sridhar and Park [26], Lamar et al. [15] and Chang and Gavish [6]). For additional references and applications see, for example, the comprehensive survey by Magnanti and Wong [19], the overview by Minoux [20], and the recent review by Crainic and Laporte [8].

In contrast to that, the published literature on network design problems for airlines is scant. Actually, we are aware of no reference treating the schedule generation problem.

Daskin and Panayotopoulos [10] present an integer program that assigns aircraft to routes in a single-hub-and-spoke network. The routes are predefined sequences of flight legs originating and terminating at the hub. They propose a Lagrangean relaxation of the problem and combine it with heuristics for converting the Lagrangean solutions into pimal feasible solutions. Barnhart and Schneur [4] describe the *express shipment service design problem*. Aircraft routes and schedules to pick up and deliver shipments have to be designed. A single hub is involved in the problem. Based on implicit column generation, a multi-label shortest path algorithm on an appropriately structured network is employed to determine new aircraft routes. Büdenbender et al. [5] present a problem similar to the one discussed here. In the context of letter mail transportation, the *direct flight network design problem* is introduced formally, and results for a practical application are given. The problem is solved using a two-phase heuristic. Rexing et al. [23] make changes to a given flight schedule by assigning time windows to each flight. After discretizing these time windows, the model is allowed to select departure times. Direct and iterative solution approaches are proposed.

3 The Mathematical Model

Roughly, an instance of the schedule generation problem (SGP) is the following: given a fleet and a set of origin-destination pairs (*OD-pairs*) with associated passenger demands, find rotations for all aircraft of the fleet such that the total profit is maximized. The next section gives a more detailed description of the problem with all its side constraints.

3.1 Problem description

The fleet consists of two sets $K = K_{\text{full}} \dot{\cup} K_{\text{part}}$ of aircraft, where K_{full} is the fleet owned by the airline and K_{part} is a set of planes that may be rented from other carriers. The planes in K_{full} are available throughout the whole planning period, and those in K_{part} are only available within specified time intervals. If utilized, these planes have to be picked up and delivered at certain airports. The rental and the (re-)positioning costs have to be provided. Each fleet consists of types of aircraft with individual characteristics as block times, minimum ground times, seat capacities and cost coefficients. Since our example sets contain only short and medium haul flights (up to 5 hours of block time), we do not consider aircraft ranges.

The set A of all airports can be partitioned into two sets of home airports $\mathcal{H}$ and those abroad $\mathcal{A}$. In our case we have 18 airports in the home country and approximately 40 airports abroad. Planes follow a *rotation*, that is in the morning, they leave an airport in $\mathcal{H}$ and have to return to such an airport in the night again. Each airport has specified opening hours, and for each flight a block time is given. In addition, aircraft rotations have to comply with a number of regulations which will be described in Section 3.2.2.

Due to the nature of the charter business, customers always book a trip starting and ending at $\mathcal{H}$ with a stay at their destination airport in $\mathcal{A}$ of one up to several weeks. This results in symmetrical demands for the OD-pairs from $\mathcal{H}$ to $\mathcal{A}$ and the corresponding return OD-pair, if one assumes, as common in the charter planning process, an even distribution of the itineraries over the season.

On the way to their destination, passengers may have to change planes or to endure a short stop-over. However, they seem to tolerate at most one intermediate stop. We assume that passengers can switch to another aircraft if the time between landing and departure is at least a given minimum connect time. Thus, passenger itineraries include either an intermediate stop in $\mathcal{H}$ or an intermediate stop in $\mathcal{A}$, maybe in combination with an aircraft switch, but not both.

The objective function is made up of several components. On the income side there are revenues for each passenger transported on an OD-pair. In contrast to regularly operating airlines, we do not have to distinguish different classes. On the cost side there are fix costs for every assignment of an aircraft to a flight leg, hire charges for rented aircraft, costs for repositioning aircraft, costs for each transported passenger (service, handling charges etc.), and penalties for passengers rejected due to capacity bottlenecks.

To summarize, the schedule generation problem is the problem to determine aircraft rotations observing operational constraints and fleet sizes, and to route passengers taking seat capacities into account, such that the combined aircraft and passenger costs are minimized, i.e. that the overall gain is maximized. Determing the aircraft routes is sufficient, but departure and arrival times are introduced as much as necessary to model the situation where passengers are allowed to switch aircraft.

We denote the classical flights, i.e. from take off to touch down and including information on departure times, by *flights*. To stress this we sometimes use the terminology of a *direct flight*. A *via flight* consists of two compatible successive direct flights $f_{1,2}$ with $\text{dest}(f_1) = \text{orig}(f_2)$, either in $\mathcal{H}$ or $\mathcal{A}$. When we describe coefficient reduction techniques and cutting planes below, we will make use of the concept of *aggregated flights*. Here, "synonymous" direct flights in our model are aggregated to a single (direct) flight. In this case it is possible that several aircraft serve such a single flight. Essentially, we can aggregate flights if the corresponding OD-pair "o $\rightarrow$ d" is not involved in any via flight.

3.2 Schedule Generation Model

We model SGP as a capacitated network design problem with additional constraints (see Section 2, especially Magnanti and Wong [19], and Minoux [20]). Compared to the general model, however, we have to observe some additional constraints: we can only introduce an arc if it is part of a feasible "virtual" route of an aircraft starting and ending in $\mathcal{H}$. Moreover, the passenger flow between two nodes can only traverse at most two flight arcs.

We remark that SGP or rather its corresponding decision problem is strongly NP-complete. This can be seen by using 3-PARTITION, which is known to be strongly NP-complete (see Garey and Johnson [12]). A proof can be found in Noltemeier [21].

3.2.1 Assumptions

Based on the problem description in Section 3.1 we make the following assumptions. Extensions of this model will be discussed in Section 7 where some of these assumptions will be dropped.

A1 (Associated home bases)
Each aircraft has an associated home base in $\mathcal{H}$ where it is parked over night. The selection of home bases can be considered as a part of the optimization and is done with respect to given airport capacity constraints.

A2 (Symmetric demands)
The demands are symmetric for each OD-pair, i.e. $\text{dem}(\text{o} \rightarrow \text{d}) = \text{dem}(\text{d} \rightarrow \text{o})$ for each day of the week. This assumption is motivated by the fact that vacations are usually booked for a multiple of a week, i.e. the departure and arrival day of the week are identical for each passenger.

A3 (Symmetric capacities)
In resemblance to Assumption **A2**, the daily aircraft rotations must provide symmetric capacities, i.e. $\text{cap}(\text{o} \rightarrow \text{d}) = \text{cap}(\text{d} \rightarrow \text{o})$ for each rotation. Thus, in this model the travelers will be transported back to their home airport one or several weeks later using the same aircraft rotation.

A4 (Via flights)
Aircraft rotations can contain two symmetric via flights (i.e., a pair of corresponding via flights) in the home country or one via flight abroad, but not both. Observe that this assumption is stronger than the restrictions on passenger itineraries as described above.

A5 (Passenger itineraries)
Passenger itineraries consist of at most two flight legs, i.e. doing a stop-over with or without switching to another aircraft is allowed only once.

Remark: By Assumptions **A2** and **A3** we do not have to consider both OD-pairs "o $\rightarrow$ d" and "d $\rightarrow$ o".

3.2.2 Aircraft Rotations

Feasible aircraft rotations must meet several requirements, including

- airport opening hours,
- earliest departure times and latest arrival times at the airports, if opening hours are not vacation-friendly (for example, very early in the morning),
- curfews (for example, by noise restrictions), and
- minimum ground times depending on departure and arrival airports of contained flights (for example for loading/unloading, refueling, and catering).

Flight durations and costs for each subfleet and for each flight are given, as well as aircraft utilization costs and the rentals for aircraft. Passenger dependent costs (as landing fees) are modeled on passenger itineraries (see the next section for details).

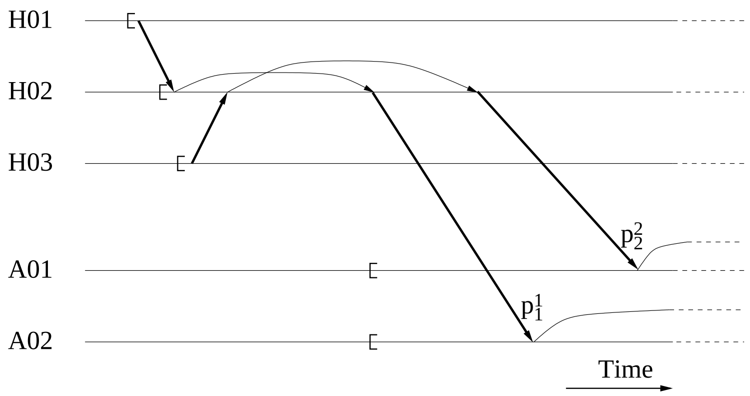

Figure 1: Illustration of aircraft rotations; with via flights in $\mathcal{H}$

Figure 1 gives an example for aircraft rotations. There are three home airports (H01, H02, and H03) and two airports abroad involved. The airports' time lines are displayed by horizontal lines. The (adjusted) airport opening hours are marked by left braces. Two rotation excerpts are shown, both with via flights in the home country. Rotation p^1_1, i.e. the first rotation (lower index) for first aircraft (upper index) starts at H01 with a flight to H02, then continuing to A02. Analogously, p^2_2 serves flights H03-H02 and H02-A01. As mentioned in Section 3.1, only the routes have to be determined. Obviously, the possibility of a passenger to get a connecting flight is determined by the exact departure and arrival times. The use of time windows in our approach is described in Section 3.2.3.

Daily aircraft rotations follow a fixed pattern. The sequence of airports in the home country and abroad is fixed. In our special case, the sequence is $H_0, H_1, A_2, A_3, H_4, A_5, H_6, A_7, H_8, H_9$, where $H_j \in \mathcal{H}$ and $A_j \in \mathcal{A}$, i.e an aircraft rotation is made up of a sequence of compatible flights $f_0, \ldots, f_8$, with $\text{dest}(f_j) = \text{orig}(f_{j+1}),\ j = 0, \ldots, 7$. This structure will be used in the "layered" column generation network, which will be described in Section 4.1 (see Figure 3). The assumptions imply that not all possible sequences of compatible flights are feasible. For example, assumption **A1** and the minimum block time for flights imply that we have an upper bound on the length of daily aircraft rotations. Thus, some of these flights may be empty, i.e., the aircraft will stay on the ground. The flights f_0, f_1, f_7, f_8, are used to model via flights in the home country and f_2 represents via flights abroad.

Aircraft rotations and passenger itineraries will be linked very closely. The revenue per passenger for the itinerary is known, as well as passenger dependent costs on aircraft (for example, service charges and landing fees). Thus, an objective coefficient combining both terms can be easily calculated.

3.2.3 Linking Aircraft Rotations and Passenger Itineraries

Considering again the example from Figure 1, Figure 2 shows six feasible passenger itineraries $i_j, j = 1, ..., 6$. Observe, that itineraries are linked with rotations. To improve the efficiency of our approach an aggregated version will be used in our implementation.

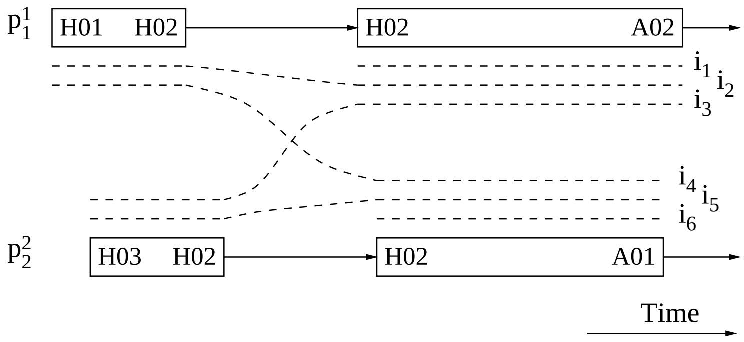

Figure 2: Illustration of link between aircraft rotations and passenger itineraries

As motivated before, departure and arrival times are important information for realistic aircraft switches. Since we are only interested in the sequence while considering all operational constraints we provide as much connecting points as possible. To describe our approach, we have to consider the two cases where via flights in the origin country or via flights abroad occur.

In the first case this is realized by "compressing" the middle part of the rotation, i.e. the flights f_1 to f_7, and placing this part on the middle of the day. Thus, the flights f_0 and f_8 are done as soon or late as possible, respectively.

In the latter case where via flights abroad are involved, we use the analog practice to create time windows between f_1 and f_2, and f_2 and f_3.

3.3 Notation and Problem Parameters

Sets:

A**:** the set of all airports, indexed by a. Indices a^- and a^+ are used for start and end airports.

K**:** the set of all aircraft, indexed by k.

Ω_d^k**:** the set of all feasible aircraft rotations for aircraft k on day d, indexed by p.

M**:** the set of all OD-pairs ("markets"), indexed by m.

I_d^m**:** the set of all passenger itineraries for OD-pair m on day d, indexed by i.

F**:** the set of all flights, indexed by (f, d) or just f, d day.

Decision variables:

θ_p^k**:** $= 1$, if rotation p flown by aircraft k is in the solution, otherwise $= 0$.

Y_{ad}^k**:** slack/surplus variables to model the position of aircraft k on airport a and day d, especially used for rental aircraft.

x_i**:** number of passengers taken on itinerary i.

s_d^m**:** the passenger spill on the OD-pair m on day d.

Data and parameters:

N**:** the length of the planning period in days, $1 \le d \le N$.

c_p^k**:** cost for using aircraft k on rotation p.

c_i**:** costs (i.e., −profit) per passenger flown on itinerary i.

cap^k**:** seat capacity of aircraft k.

n_d^k**:** $= 1$, if aircraft k is available on day d, otherwise $= 0$.

r_{ad}^k, s_{ad}^k**:** $\in \{-1, 0, 1\}$, lower and upper bounds on the supplementary variables Y_{ad}^k.

dem_d^m**:** the demand for the OD-pair m on day d.

χ_{fp}^k**:** the multiple of the aircraft capacity cap^k provided in rotation p for the flight f, maybe 0.

δ_{fp}, δ_{fi}**:** $= 1$, if aircraft rotation p or passenger itinerary i contains the flight f, respectively, otherwise $= 0$.

δ_{ip}**:** $= 1$, if the passenger itinerary i and the aircraft rotation p are sharing a flight, otherwise $= 0$.

$\mathrm{orig}(i), \mathrm{dest}(i), \mathrm{orig}(p), \mathrm{dest}(p), \mathrm{orig}(f), \mathrm{dest}(f)$**:** origin and destination of passenger itinerary i, aircraft rotation p, and flight f, respectively.

3.4 Path-based Mixed Integer Programming Formulation

$$\text{(SGM =) Minimize} \quad \sum_{1 \le d \le N} \left(\sum_{k \in K} \sum_{p \in \Omega_d^k} c_p^k \theta_p^k + \sum_{m \in M} \sum_{i \in I_d^m} c_i x_i \right) \tag{1}$$

subject to

$$\sum_{m \in M} \sum_{i \in I_d^m} \delta_{fi} x_i \le \sum_{k \in K} \mathrm{cap}^k \sum_{p \in \Omega_d^k} \chi_{fp}^k \theta_p^k \qquad \forall\, (f,d) \in F \qquad (\alpha_{(f,d)}) \tag{2}$$

$$\sum_{p \in \Omega_d^k \setminus \{0\}} \theta_p^k \le n_d^k \qquad \forall\, k \in K,\ \forall\, 1 \le d \le N \qquad (\beta_{kd}) \tag{3}$$

$$\sum_{\substack{p \in \Omega_{d+1}^k \\ \mathrm{orig}(p)=a}} \theta_p^k - \sum_{\substack{p \in \Omega_d^k \\ \mathrm{dest}(p)=a}} \theta_p^k - Y_{ad}^k = 0 \qquad \forall\, k \in K,\ \forall\, a \in A,\ \forall\, 1 \le d \le d+1 \le N \qquad (\gamma_{kda}) \tag{4}$$

$$\sum_{\substack{p \in \Omega_1^k \\ \mathrm{orig}(p)=a}} \theta_p^k - \sum_{\substack{p \in \Omega_N^k \\ \mathrm{dest}(p)=a}} \theta_p^k - Y_{aN}^k = 0 \qquad \forall\, k \in K,\ \forall\, a \in A \qquad (\gamma_{kNa}) \tag{5}$$

$$r_{ad}^k \le Y_{ad}^k \le s_{ad}^k \qquad \forall\, k \in K,\ \forall\, a \in A,\ \forall\, 1 \le d \le N \tag{6}$$

$$\sum_{i \in I_d^m} x_i + s_d^m = \mathrm{dem}_d^m \qquad \forall\, m \in M,\ \forall\, 1 \le d \le N \qquad (\epsilon_{md}) \tag{7}$$

$$\theta_p^k \text{ binary} \qquad \forall\, k \in K,\ \forall\, p \in \Omega^k \tag{8}$$

$$x_i \ge 0 \qquad \forall\, m \in M,\ \forall\, i \in I^m \tag{9}$$

$$s_d^m \ge 0 \qquad \forall\, m \in M,\ \forall\, 1 \le d \le N \tag{10}$$

The objective function (1) minimizes the total cost, which is the sum of aircraft utilization costs and costs on passenger itineraries. The link between provided aircraft capacities and transported passengers on itineraries is done by constraint set (2). Aircraft availability is ensured by constraints (3). Constraint sets (4) and (5) define classical flow conservation constraints for the aircraft rotations. For modeling complex situations where aircraft are only available partially, the slack and surplus variables Y^k_{ad} are introduced. Lower and upper bounds on these supplementary variables are enforced by constraints (6). The relationship between the number of transported passengers for an OD-pair and the corresponding spill is given by (7). Constraints (8) impose binary values for the path variables. Finally, passenger and spill variables are ensured to be non-negative by constraint sets (9) and (10).

3.5 Overview of the Algorithm

Depending on the number of possible aircraft rotations, we propose to use a branch-and-cut or a branch-and-cut-and-price algorithm to solve the problem.

In the given instances, there is only a moderate number of number of feasible rotations when using aggregation. Thus working on the explicit integer linear program is possible. For larger instances, column generation will become necessary. The linear relaxation of SGM, i.e. (1) – (10) provides only a weak relaxation. We will develop cutting planes valid throughout the branch-and-bound tree to tighten the relaxation.

4 Column Generation Subproblems

For large schedule generation instances, i.e. for instances with a large number of airports in $\mathcal{H}$ and $\mathcal{A}$, and many OD-pairs with positive demands, the number of feasible aircraft rotations can become very large and thus can not be handled explicitely anymore. Also, relaxing all or some of Assumptions **A1**, **A3**, and **A4** drastically increases the number of aircraft rotations. In this case we propose to apply a branch-and-price technique utilizing constrained shortest path problems for generating new rotations dynamically, and thus to solve the problem. We refer the reader, beside many other references, to Lasdon [16] for a detailed introduction, and to Barnhart et al. [3] and Desaulniers et al. [11] for recent articles on modeling and solving problems using branch-and-price. For the given instances and when observing all assumptions, the number of aircraft rotations is of moderate size, and thus we work on the explicitely enumerated set of all rotations in our computational study.

In Section 4.1 we describe the aircraft rotation subproblem, while Section 4.2 discusses the passenger itinerary subproblem.

4.1 Aircraft Rotation Subproblem

Let (θ, x) be the solution vector of the current restricted master problem (RMP), and let $(\alpha, \beta, \gamma, \epsilon)$ be its associated dual solution vector. Then, the reduced cost $\bar{c}^k_p$ of an aircraft rotation $p \in \Omega^k_d$ for aircraft k on day d, starting and ending at airport $a^-, a^+ \in A$, respectively, is

$$\bar{c}^k_p = c^k_p + \sum_{(f,d) \in F} \mathrm{cap}^k \chi^k_{fp} \alpha_{(f,d)} - \beta_{kd} - \gamma_{kda^-} + \gamma_{kda^+}, \tag{11}$$

using the convention $\gamma_{k0a^-} = \gamma_{kNa^-}$ for the period wraparound. Observe, that SGM (i.e. (1) - (10)) and (11) allow different airports $a^- \neq a^+$ for departure in the morning and arrival in the night, and thus is more general. We will discuss this as a possible extension in Section 7. For each aircraft and each licensed home base a single network is defined. Nodes represent airport-layer pairs according to the aircraft rotation pattern. Furthermore, source and sink nodes are added and associated with the home base. Two different arc types are used. Arcs adjacent to the source or the sink node are utilization arcs. They are used to keep track of the number of utilized aircraft.

Arcs connecting airport-layer nodes represent flight activities. Basic restrictions such as aircraft ranges and curfews can be considered directly in the network, because only admissible flight arcs are inserted.

The aircraft rotation subproblem now is the problem of finding an aircraft rotation that satisfies all operational constraints, and prices out to have negative reduced cost. This problem can be cast as the problem of finding shortest paths in the network defined above. There is a one-to-one correspondance between feasible paths in the network and admissible aircraft rotations. Further, the reduced cost components in (11) can be transfered to the arcs of the network, such that the paths cost equals the reduced cost. To comply with the station opening hours and aircraft rotation pattern, the networks can be solved by utilizing a resource-constrained shortest path algorithm. For example, each arc in the network uses up an amount of the resource "time" (e.g., flying, ground and waiting time). The comsumption of this resource is checked at each node in the network. Time intervals represent the associated airport opening hours. Beside other possibilities, this network can be used to enumerate all admissible paths. They can be stored in column pools, and reduced costs can be calculated using (11). Figure 3 illustrates this network structure. An

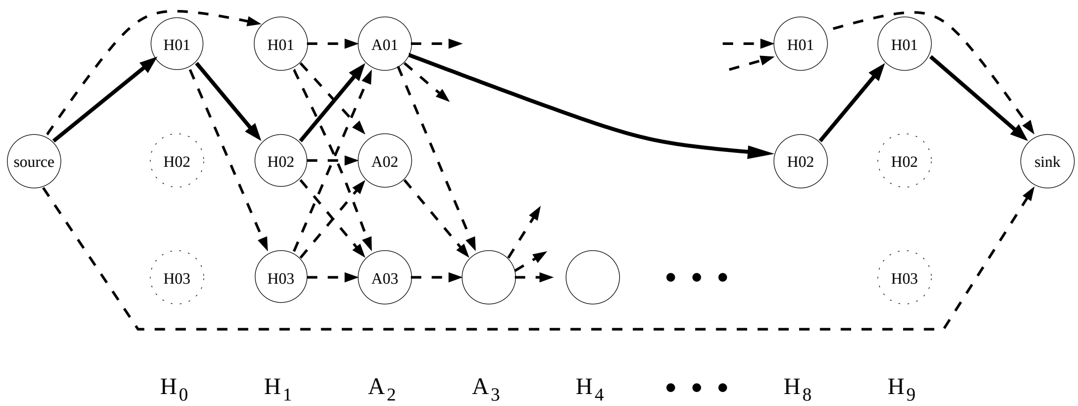

Figure 3: Network structure for the aircraft rotation subproblem, home base H01

excerpt for a network for an aircraft with home base H01 is shown. A rotation serving a via flight between home airports H01 and H02 (see layers 0 and 1, and 8 and 9), and serving flights H02–A01–H02 to airport A01 abroad is shown by solid lines. This rotation omits nodes in layers three to seven. Other flights in the network are indicated by dashed lines. Between layers there is not necessarily a complete bipartite graph. For example, airport-layer nodes H01 in layers 0 and 1 are not connected, because this only represents staying on the ground, and some arcs may missing caused by curfews. A special arc for not utilizing the aircraft is also shown and connects the source node with the sink node.

4.2 Passenger Itinerary Subproblem

Again, let $(\alpha, \beta, \gamma, \epsilon)$ be the current dual solution vector. The reduced cost $\bar{c}_i$ of an passenger itinerary $i \in I_d^m$ for OD-pair m on day d is

$$\bar{c}_i = c_i - \sum_{(f,d)\in F} \delta_{fi}\alpha_{(f,d)} - \epsilon_{md}. \tag{12}$$

The number of passenger itineraries is small, because only one aircraft switch is allowed by Assumption **A5**. Thus, we propose to explicitly enumerate all itineraries, and compute their reduced costs directly, when necessary.

5 Obtaining Integer Solutions

Solving the linear relaxation of the formulation as stated in Section 3.4 directly leads to high fractionality in the aircraft rotation variables θ_p^k. In this section, we will identify the set of constraints of SGM causing the fractionality and being responsible for the weak linear relaxation. We propose coefficient reduction and cutting planes to tighten the linear relaxation bound. Further, we will discuss a depth-first heuristic and branching strategies for obtaining integer solutions.

5.1 Causes of Fractionality

This fractionality is caused by constraint set (2) of SGM,

$$\sum_{m \in M} \sum_{i \in I_d^m} \delta_{fi} x_i \leq \sum_{k \in K} \mathrm{cap}^k \sum_{p \in \Omega_d^k} \chi_{fp}^k \theta_p^k \qquad \forall\, (f, d) \in F.$$

For the analysis, we take a look on a single OD-pair. We ignore the remainder of the involved aircraft rotations, and the aircraft utilization and flow conservation constraints, since they only play a minor role.

OD-pair	Demand	Rev./Pass.
H01–A01	144	DEM -360
H02–A01	174	DEM -365
H03–A01	162	DEM -380

Table 1: Demand data

Path Variable	Operating Costs	Rotations, providing a capacity of 174 seats per flight				
p_1	DEM 45,000	H01	A01	H01		
p_2	DEM 47,000	H02	A01	H02		
p_3	DEM 53,000	H03	A01	H03		
p_4	DEM 77,000	H01	A01	H02	A01	H01
p_5	DEM 83,000	H01	A01	H03	A01	H01
p_6	DEM 85,000	H02	A01	H03	A01	H02

Table 2: Operating costs and rotations

Assume, there are three OD-pairs H01–A01, H02–A01, and H03–A01, and only one subfleet (the Airbus A320 with an capacity of 174 seats) involved in this problem. Via flights are not allowed. Using the demand and revenue data of Table 1 and the aircraft rotations, operational costs and capacities of Table 2, the optimal solution to the relaxed problem of SGM is achieved by $\theta_{p_4} \sim 0.448$, $\theta_{p_5} \sim 0.379$, $\theta_{p_6} \sim 0.552$ and all other path variables =0, with an objective function (cost) value of DEM -64,013. The capacity provided is equal to the demand and all passengers are transported. The optimal integral solution is achieved by $\theta_{p_1} = \theta_{p_6} = 1$ and all passengers transported, with an objective function value of DEM -46.910, thus giving an integrality gap of ~26.7%.

5.2 Coefficient Reduction

A method to tighten the linear relaxation bound and thus to improve the solution behavior is to reduce each coefficient of the aircraft rotations in constraint set (2) to be not greater than the left hand side (LHS). Obviously, no integral solutions are cut off by this practice. Usually, the

Path Variable	Operating Costs	Rotation and provided capacities (after preprocessing)
p_1	DEM 45,000	H04 (144) A02 (144) H04
p_2	DEM 47,000	H05 (174) A02 (174) H05
p_3	DEM 53,000	H06 (162) A02 (162) H06
p_4	DEM 77,000	H04 (144) A02 (174) H05 (174) A02 (144) H04
p_5	DEM 83,000	H04 (144) A02 (162) H06 (162) A02 (144) H04
p_6	DEM 85,000	H05 (174) A02 (162) H06 (162) A02 (174) H05

Table 3: Operating costs, capacities, and rotations

demand of passengers using a flight is greater than each aircraft capacity, particularly if passenger itineraries and via flights are involved. Using again the demand data of Table 1, the provided seat capacities for the above example become as summarized in Table 3. The optimal solution to the relaxed problem is achieved by $\theta_{p_4} = 0.503067$, $\theta_{p_5} = 0.496933$, $\theta_{p_6} = 0.496933$, with an objective function value of DEM -54,6891, and an integrality gap of $\sim$14.2%. Again, all passengers are transported.

5.3 'Min-Cover' Inequalities

Lifted cover inequalities were introduced by Crowder et al. [9] and applied to pure binary programs. Padberg et al. [22] and van Roy and Wolsey [27] extended these inequalities to generalized flow cover inequalities for mixed-binary programs. Our cutting planes are related to the cuts described in these references. We will compare the results of our computational experiments with results obtained by the CPLEX mixed-integer program solver (see [1]) and MINTO (see Savelsbergh and Nemhauser [25]) in Section 6.2.4. In these solvers cuts following the line of Crowder et al. are used.

To describe the application of min-cover inequalities to the SGP, we temporarily assume, that no via flights are involved in the planning process, i.e., no itineraries consisting of two flights are given and thus demands can only be satisfied by direct flights. In the following we consider a specific aggregated direct flight *locally*, i.e. we ignore the remaining part of the rotations and the aircraft utilization constraints.

Now assume that coefficient reduction has already been performed and consider a fixed aggregated flight $(f, d) \in F$ with a demand of the corresponding OD-pair of *dem*. The passenger itinerary and aircraft rotation linking constraint (2) can be rewritten as follows. The LHS reduces to a single passenger variable, say x_i. The right hand side (RHS) can be expanded by using the actual provided capacity $\overline{\text{cap}}_f^k(p)$ of each aircraft rotation p, incorporating the multiplier χ_{fp}^k and the reduced coefficients. Thus,

$$x_i \leq \sum_{k \in K} \sum_{p \in \Omega_d^k} \overline{\text{cap}}_f^k(p)\, \theta_p^k. \tag{13}$$

Let $c_1 < \ldots < c_n$ be the different (non-zero) capacities $\overline{\text{cap}}_f^k(p)$ in (13). Observe, that there may be more different capacities c_j than aircraft capacities. The aircraft rotations can now be partitioned according to their capacities, i.e.

$$\overline{\Omega}_d^j = \left\{(p, k) \mid k \in K,\ p \in \Omega_d^k,\ \overline{\text{cap}}_f^k(p) = c_j\right\}$$

for $1 \leq j \leq n$. Introduce new variables y_j counting the number of aircraft serving (f, d), each providing a corresponding capacity of c_j, i.e.

$$y_j = \sum_{(p,k) \in \overline{\Omega}_d^j} \theta_p^k \quad (\in \mathbb{N}).$$

The variables y_j are implicitly integral. Further, we use the variable sp to denote the number of passengers that are spilled on this flight. Obviously, the following cover inequality must hold:

$$c_1 y_1 + \ldots + c_n y_n \geq \text{dem} - sp. \tag{14}$$

Let

$$X = \{(y, sp) \in \mathbb{N}^{n+1} \mid \text{Inequality (14) holds for } (y, sp)\}$$

be the set of all feasible integer configurations, and $\mathcal{P} = \text{conv}(X)$ the convex hull of X. The local subproblems occurring in SGM are of moderate size, since the number of different aircraft seat capacities is limited. Thus, we can calculate all facets of $\mathcal{P}$ in advance during a preprocessing step. The computation of the facets is performed using e.g. PORTA [7]. The inequalities corresponding to the facets of $\mathcal{P}$ are valid for SGM (in absence of the neglected constraints). Clearly, they are valid for the complete SGM, too, but not necessarily facet inducing anymore.

When via flights and passenger itineraries are involved, min-cover inequalities can be used, too, although the situation becomes more complicated. In contrast to the case without via flights, the variables $y_j, 1 \leq j \leq n$ correspond to the number of aircraft with a certain capacity c_j serving a direct flight or a follow-on. A follow-on is a sequence of two compatible successive direct flights. In our case, we consider follow-ons completely in $\mathcal{H}$ and follow-ons completely in $\mathcal{A}$ (i.e. the via flights as described above), in addition to the direct flights. However, the number of allowed via flights is small in our test instances, and the local min-cover facets can be calculated explicitly, see Section 6.2.

Observe that for our computational study all facets to the (local) min-cover polyhedron are calculated explicitly. The computational effort for explicitly computing the facets depends on the demand *dem*, the number of different aircraft seat capacities $c_1, \ldots, c_n$, and the number of allowed via flights. If computation time exceeds justifiable limits, "aggregated" capacities can be used. If, for example, an aircraft rotation with a capacity of c_l seats serves a direct flight twice, this can be seen as an aircraft with seat capacity $c_j = 2c_l$ serving this direct flight once. This may increase the number of different capacities, but leads to deeper cuts than expressing this as $2c_l$. Thus, the expanded version should be used whenever possible.

5.4 A Depth-First Heuristic

To obtain good feasible solutions quickly, we use a search heuristic. The heuristic fixes aircraft rotations to 1 in a depth-first manner, and can be applied at every node of the branch-and-bound tree. Obviously, every aircraft rotation solution taking aircraft utilization and flow conservation constraints into account yields a feasible solution. Thus, we do not have to employ column generation while applying the heuristic.

❶ Fix one (or a few) aircraft rotation variable(s) θ_p^k with

$$\text{ROUND\_THRESHOLD} \leq \theta_p^k < 1$$

to 1. ROUND_THRESHOLD is a value near to 1, and gives the threshold for the variable to be a candidate for rounding to 1. Usual values chosen are 0.85 or 0.95.

❷ Resolve the LP without employing column generation.

❸ Fix aircraft rotation variables θ_p^k with high (positive) reduced costs exceeding a given limit to 0.

❹ If the current solution is integer, update the best feasible solution.

❺ If the current objective function value is smaller than best known feasible solution, goto ❶.

❻ Make a few backtracking steps using different aircraft rotation variable(s) in ❶,
or: start a complete new search with ❶,
or: stop the heuristic.

This heuristic can be extended for weekly problems in a straightforward way. A weekly rotation for an aircraft in K_{full} has to be found and fixed. This rotation consists of aircraft rotation variables θ_p^k for every day of the week. Aircraft from K_{part} are handled similar. The efficiency of this heuristic will be stated in Section 6.2.

5.5 Branching

We have applied different branching strategies in our computational experiments to obtain integer solutions. Although the standard branching rule on variable dichotomy makes the pricing problem more complicated, the empirical results are worth mentioning. Thus, besides more sophisticated branching rules it will be described here, too.

5.5.1 Variable Dichotomy

The standard branching rule for Mixed Integer Programming is not directly applicable if column generation is used. In the case that variables are set to 0, the pricing problem must incorporate these decisions and not return to this variable again. This leads essentially to finding the $l+1$-best variable in the subproblem, if l variables are already set to 0. In addition, setting variables to 0 does not divide the search space in two equal-sized parts, which is usually preferred. Nevertheless, setting variables only to 1 is closely related to the depth-first heuristic (when working on a fixed set of columns) and yields good solutions.

5.5.2 Number of used Aircraft

If the number n' of aircraft used on day d in the current relaxation, i.e.

$$n' = \sum_{k \in \bar{K}} \sum_{p \in \Omega_d^k} \theta_p^k \tag{15}$$

with $\bar{K} = K$, is fractional, branches that require this number to be $\leq \lfloor n' \rfloor$ and $\geq \lceil n' \rceil$ can be created. This rule can be applied to subsets of K, too. For example, the set K of all aircraft can be partitioned into subfleet groups $K = K_1 \dot{\cup} \cdots \dot{\cup} K_l$, each containing the same aircraft type. Thus, the number of aircraft used of a given type (n' in (15) with $K = K_i$ for a subfleet i) is required to be $\leq \lfloor n' \rfloor$ and $\geq \lceil n' \rceil$, respectively.

5.5.3 Number of Aircraft on Direct Flights

Consider an aggregated direct flight (f, d) on a fixed day d. This flight may be flown by several aircraft rotations involving several aircraft in the current relaxation. Obviously, the number of aircraft serving this flights has to be integral. Thus, for a fractional value

$$r' = \sum_{k \in K} \sum_{\substack{p \in \Omega_d^k \\ (f,d) \in p}} \theta_p^k,$$

branches forcing the number to be $\leq \lfloor r' \rfloor$ and $\geq \lceil r' \rceil$ may be created. Analog to 5.5.2, this rule can be applied to subfleet groups, too.

5.5.4 Number of Aircraft on Follow-Ons

Consider two compatible successive direct flights (f_j, d), $\text{dest}(f_1) = \text{orig}(f_2)$, $j = 1, 2$. As introduced in Section 5.3, the combination $f_1 \to f_2$ is called a follow-on. A Ryan/Foster-like branching rule (Ryan and Foster [24]) on follow-ons is frequently used for routing and scheduling type problems (see, for example Desaulniers et al. [11] and Barnhart et al. [28]). In most formulations, a task has to be covered exactly once. Thus, the flow on this follow-on is forced to be 0 or 1, respectively.

In our formulation, several aircraft may serve direct flights and follow-ons. We propose an extension of this rule applicable to the SGP. If the number of aircraft r' serving a follow-on is fractional with

$$r' = \sum_{k \in K} \sum_{\substack{p \in \Omega_d^k \\ (f_1 \to f_2) \in p}} \theta_p^k,$$

two branches $\leq \lfloor r' \rfloor$ and $\geq \lceil r' \rceil$ as described above are created.

6 Computational Results

To prove the efficiency of our approach, we provide experimental results on data sets received from a major charter airline. These instances describe a typical summer week. There are 18 home airports, and ~40 destination airports. All of them are reachable by short and medium haul flights.

We would like to remark, that for the results given in this paper *perturbed data sets* have been used. The revenues given do not correspond to revenues of the airline! Besides, it seems that the charter airline used a different objective function and maybe some additional constraints. Thus, the improvements of our solutions compared to the hand solutions provided by the charter airline seem to be non realistic.

All computational results are obtained on a customary 333 MHz Pentium-II personal computer equipped with 256 MByte RAM. All run-times shown in the tables are in seconds, and do *not* include the time for reading the input files and preparing the initial data (such as kernel LP and networks).

The `BnP` (Branch-&-Price) library developed at ZAIK is used as a framework. The primal und dual simplex algorithms of CPLEX [1] are used for solving all LP relaxations, and are called as a subroutine from `BnP`.

6.1 Data Sets

Name	S98-Mo	S98-Tu	S98-We	S98-Th	S98-Fr	S98-Sa	S98-Su
Num. H airports	16	11	16	13	14	16	15
Num. A airports	9	10	11	11	11	15	12
Num. of OD-pairs	47	48	60	54	55	89	63
Sum of demands	10.011	10.071	12.220	10.768	14.707	19.601	18.124
Num. of AC types	6	7	7	7	6	11	9
Num. of AC	29	30	30	30	33	64	53
$\lvert K_{\text{full}} \rvert + \lvert K_{\text{part}} \rvert$	26+3	26+4	26+4	26+4	26+7	26+38	26+27

Table 4: Characteristics of "Summer 1998" data set (single days)

The characteristics of our test instances are presented in Table 4. The columns show each day of a standard summer week (Summer 1998), named S98-Mo through S98-Su. The rows *Num. H airports* and *Num. A airports* give the number of home airports and airports abroad, respectively. The next both rows indicate the demand structure. The number of OD-pairs with nonzero demand, and the overall sum of passengers are given. In the last block, *Num. of AC types* shows the number of different subfleets involved. Finally, *Num. of AC* gives the number of available aircraft. The both components indicate the number of own aircraft ($|K_{\text{full}}|$) and the number of aircraft the company may rent from other carriers ($|K_{\text{part}}|$).

The complete week S98-Week makes up one further instance. It consists of 18 home airports, 38 abroad and of 266 OD-pairs. The sum of demands and the number of subfleets and available AC can be obtained by summing the characteristics of the involved days.

6.2 Results

The results had been carried out in various test series. Decisions that have to be done include whether via flights are involved (and if, where they are allowed) or not. Also, the distribution of aircraft at the beginning of the planning horizon may be predetermined or considered as part of the optimization. They interact with the airport capacities, since only limited space for parking aircraft overnight is available. Further, they must observe the defaults of the airline. The airline provided hand-made schedules to us and the aircraft distributions from these solution will be used. The number of spilled passengers plays an important role, too. Although, usually the market data give overbooked demands, the passenger spill should not exceed certain limits. Thus, restricting the spill on certain OD-pairs or the overall spill, maybe in combination with penalty costs, is possible.

In this section we give computational results for different scenarios. We split the results into four parts. Sections 6.2.1 and 6.2.2 deal with the SGP using a predetermined or free aircraft distribution, respectively, but with no via flights allowed. Results for the weekly problem can be found in section 6.2.2, too. Section 6.2.3 discusses the case with via flights involved. In Section 6.2.4, we compare the results achieved by our optimizer with the results obtained by CPLEX (see [1]) and MINTO (see Savelsbergh and Nemhauser [25]).

6.2.1 Results without via flights, fixed distribution of AC

Table 5 summarizes the results for each day of the "Summer 1998" instance. In this instances the aircraft distribution was predetermined using the distribution in the given hand-solution. The first block of this table gives information on the solution of the branch-and-bound root node. The row *LB w/o cuts* shows the relaxation lower bound (LB) after solving SGM without any additional cuts. The number of cutting plane iterations and inserted min-cover cuts is given next. The LB of the tightened relaxation is given in the next row. The bound improvement is shown in the row *Bound impr.* and is calculated by $\left(\frac{\text{LB w/o cuts} - \text{LB w/cuts}}{\text{LB w/o cuts}}\right)$. The row *CPU* displays the used CPU time in seconds. The next blocks show information on the first and best feasible solutions found. The CPU time was limited to 1 hour. *B&B node* gives the number of the branch-and-bound node in which this solution was found. Next, again information on the cutting plane process are shown. New cuts were only generated for branch-and-bound nodes with depth ≤ 3, and the numbers refer to all nodes but the root node. Only successful runs of the separation routine, i.e. runs finding violated cuts, were counted in *Cut iter.*. For all instances the depth-first heuristic applied in the root node yields a (first) feasible solution. Thus, there are no branch-and-bound and cutting plane information for the "First feas." block. The objective value and the gap (giving the quality guarantee) are provided next. For a feasible solution with objective value "Obj. feas." the latter is calculated by $\left(\frac{\text{LB w/cuts} - \text{Obj. feas.}}{\text{LB w/cuts}}\right)$. Next, the accumulated CPU time is shown. For a rough comparison of our obtained solutions with the given hand-solutions, the number of (overall) spilled passengers and the number of utilized aircraft are given. The last block provides information on the hand-solution. Although we do not allow via flights in these runs, the airline solution contains via flights, and its evaluation was elaborated incorporating these via flights. A comparision of our obtained solutions (without via flights) with the given hand-solution (containing via flights) shows, that our approach reduces the number of used aircraft, but increases the number of spilled

Name	S98-Mo	S98-Tu	S98-We	S98-Th	S98-Fr	S98-Sa	S98-Su
Root node							
LB w/o cuts	-1,230,556	-1,186,034	-1,242,441	-1,381,389	-1,592,853	-2,178,304	-2,118,376
Cut iter.	4	4	2	5	7	11	9
Cuts inser.	38	29	27	45	75	144	130
LB w/cuts	-1,146,974	-1.120.285	-1,165,879	-1,274,329	-1,505,859	-2,057,952	-2,003,195
Bound impr.	6.79%	5.54%	6.31%	7.75%	5.46%	5.53%	5.43%
CPU	4	5	15	7	27	94	139
First feas.							
Obj.	-1,117,702	-1,099,686	-1,163,920	-1,240,945	-1,482,556	-1,962,334	-1,904,021
Gap	2.55%	1.84%	0.17%	2.61%	1.55%	4.65%	4.95%
CPU	6	7	19	9	33	115	230
Spilled Pass.	1045	1,053	2,053	1,050	1,424	1,878	1,804
Num. used AC	22	23	23	24	28	49	39
Best feas.							
B&B node	53	2	7	7	7	15	17
Cut iter.	2	2	2	0	1	1	0
Cuts inser.	3	2	3	0	1	4	0
Obj.	-1,136,116	-1,119,281	-1,163,921	-1,273,197	-1,486,976	-2,051,895	-1,947,594
Gap	0.95%	0.09%	0.17%	0.09%	1.25%	0.29%	2.78%
CPU	123	11	59	34	407	487	2,168
Spilled Pass.	808	939	2,053	717	1,394	1,226	1,186
Num. used AC	22	23	23	24	28	52	40
Airl. Sol.							
Spilled Pass.	553	445	852	617	631	709	598
Num. used AC	29	30	30	30	33	64	53

Table 5: Results for "Summer 1998" (fixed distribution of aircraft, w/o via flights)

Name	S98-Mo	S98-Tu	S98-We	S98-Th	S98-Fr	S98-Sa	S98-Su
Fixed AC distribution							
Constraints	197	200	224	212	213	320	257
Variables	3,898	2,665	11,257	5,883	7,340	17,231	13,401
Nonzero's	19,119	11,824	58,418	28,884	38,427	87,732	70,336
Cuts	644	2,094	4,836	1,151	3,030	3,254	6,376

Table 6: Statistics on "Summer 1998" data set (fixed distribution of aircraft, w/o via flights)

passengers. The objective value is improved by about 21% in average, but this improvement must be handled very carefully as discussed above.

The characteristics of the aggregated linear programs and the number of precalculated min-cover inequalities are shown in Table 6. For this computational study, fixed costs for transporting a passenger are independent of the used aircraft type. Using the more detailed cost components will of course not influence the number of min-cover inequalities but the required times for solving the linear programs. The size of the linear programs is moderate and no dynamic column generation is needed. Clearly, if more flights per rotation are allowed or more airports are involved this number will increase significantly. Min-cover cuts are computed and stored in a database whenever a new "local" problem occurs. The running-time required to calculate all cuts necessary for the problems in this paper is less than 2 hours.

These results demonstrate that "mathematically" convincing solutions to the SGP could be obtained using the proposed model. The lower bounds were improved by 6.12% in average and the average gap amounts to 0.80%. In these runs neither the passenger spill was bounded (global or on OD-pairs) nor penalty costs for spilled passengers were applied. A drawback of these solutions is the high number of spilled passengers. It ranges between $1,16$ and $2,41$ times the corresponding spill in the given hand-solutions. The demands are usually made up of two or three different components, *FIX*, *ProRata* and *I*. The component *FIX* represents the largest part. In average, this is more than 85%. It contains especially seats already sold to tour operators. In contrast, the components *ProRata* and *I* are sold directly. Thus, spilling passengers from *FIX* should be avoided. Table 7 summarizes the results for all days incorporating limited and penalized spill. No penalty costs apply for spilling passengers from *ProRata* and *I* but a linearization of a quadratic cost function for spilling passenger from *FIX* is charged. The objective value given does not reflect the penalty cost and thus is comparable to our other results. These results are not achieved in a single run but are the outcome of an iterative and interactive process in which penalty costs and hard limits have been set and readjusted a (very) few times. In an industrial use of our system the skills of scheduler are still necessary, yet. The number of used aircraft increased compared to Table 5 but is still a reduction compared to the airline hand solution. Also, renting aircraft is usually due to a long term contract and so it is not the most important aim to reduce the number of aircraft used. Further, the number of spilled passengers could be decreased compared to both our previous solution and the hand solutions.

Name	S98-Mo	S98-Tu	S98-We	S98-Th	S98-Fr	S98-Sa	S98-Su
Solutions							
Obj.	-1,132,803	-1,083,427	-1,125,464	-1,271,571	-1,444,010	-1,881,585	-1,903,347
Gap	1.24%	3.29%	3.47%	0.22%	4.11%	8.57%	4.98%
Spilled Pass.	476	409	773	604	631	689	583
Num. used AC	23	25	26	25	31	54	42

Table 7: Results for "Summer 1998" (fixed distribution of aircraft, with bounded spill, w/o via flights)

6.2.2 Results without via flights, free distribution of AC

Table 8 shows the results for a predetermined aircraft distribution for rented aircraft only, as explained above. Own aircraft are allowed to reposition. For the two data sets "S98-Tu" and "S98-We" the solution to the relaxation is feasible, too. For most of the data sets, the number of spilled passenger is smaller compared to the solution with fixed aircraft distribution. The lower bounds were improved by 5.67% in average and the average gap amounts to 0.84%.

Table 9 summarizes the results for the complete week instance "Summer 1998" without via flights, where own aircraft are allowed to reposition. The first block of this table gives information on the solution of the branch-and-bound root node, the second shows information on the first and

Name	S98-Mo	S98-Tu	S98-We	S98-Th	S98-Fr	S98-Sa	S98-Su
Root node							
LB w/o cuts	-1,332,149	-1,209,941	-1,286,956	-1,411,771	-1,653,388	-2,222,227	-2,155,338
Cut iter.	4	8	4	4	6	9	12
Cuts inser.	43	36	40	53	78	145	139
LB w/cuts	-1,243,407	-1,149,911	-1,199,076	-1,305,578	-1,559,230	-2,109,345	-2,047,228
Bound impr.	6.66%	4.96%	6.83%	7.52%	5.69%	5.08%	5.02%
CPU	3	8	23	9	28	77	141
First feas.							
Obj.	-1,229,153	-1,149,910	-1,199,075	-1,245,924	-1,327,036	-2,030,912	-1,905,485
Gap	1.15%	0%	0%	4.56%	14.8%	3.72%	6.92%
CPU	4	9	27	12	31	177	183
Spilled Pass.	566	890	1,686	1,232	2,010	1,351	868
Num. used AC	23	24	23	22	28	48	41
Best feas.							
B&B node	194	0	0	19	12	42	10
Cut iter.	0	0	0	0	1	1	0
Cuts inser.	0	0	0	0	14	1	0
Obj.	-1,232,572	-1,149,910	-1,199,075	-1,302,818	-1,539,635	-2,079,489	-2,003,694
Gap	0.87%	0%	0%	0.21%	1.26 %	1.42%	2.13%
CPU	305	9	27	103	189	3,522	676
Spilled Pass.	534	890	1,686	761	841	1,157	887
Num. used AC	23	24	23	23	30	52	41

Table 8: Results for "Summer 1998" (free distribution of aircraft, w/o via flights)

best feasible solution found. The columns *Sp. Pass.* and *Num. AC* give the number of spilled passengers and the sum of the number of utilized aircraft, respectively.

	LB w/o cuts	Cut iter.	Cuts inser.	LB w/cuts	B. impr.		
Root node	-11,254,926	12	620	-10,566,445	6.12%		
	B&B node	Cut iter.	Cuts inser.	Obj.	Gap	Sp. Pass.	Num. AC
First feas.	–	–	–	-10,015,498	5.21%	9,042	204
Best feas.	5	0	0	-10,355,282	2.00%	9,480	204

Table 9: Results for complete week "Summer 1998" (free distribution of aircraft, w/o via flights)

A comparision of our best feasible (weekly) solution to the given hand-solution showed, that we spill about 5075 passengers more than the airline, but reduce the sum of used aircraft from 269 to 204 and increase the objective value about 23%.

Remark, that the best feas. solution above achieves a better objective value and reduces the number of spilled passengers compared to the joined solution with fixed aircraft distribution (see Table 5).

6.2.3 Results with via flights

Table 10 summarizes the results for the "Summer 1998" instance with via flights included. The aircraft distribution was predetermined using the distribution in the given hand-solution. The set of considered via flights was restricted to a given (from the airline) set of allowed via flights.

The lower bounds were improved by 5.97% in average and the average gap amounts to 0.54%. Since the number of possible flights increases by including via flights, the objective value of the relaxation of the SGM with additional via flights without any additional cuts and the objective value of the tightened relaxation are better than the corresponding objective values without via flights (see Table 5). All but one feasible solutions include via flights. This seems to happen

Name	S98-Mo	S98-Tu	S98-We	S98-Th	S98-Fr	S98-Sa	S98-Su
Root node							
LB w/o cuts	-1,275,794	-1,197,550	-1,274,681	-1,415,317	-1,627,163	-2,223,668	-2,133,967
Cut iter.	4	4	5	4	5	8	9
Cuts inser.	39	29	44	47	72	123	146
LB w/cuts	-1,188,105	-1,129,078	-1,196,086	-1,314,771	-1,538,560	-2,102,624	-2,016,670
Bound impr.	6.88%	5.72%	6.17%	7.10%	5.45%	5.44%	5.50%
CPU	16	8	89	21	59	235	328
First feas.							
Obj.	-1,179,838	-1,129,078	-1,193,168	-1,314,247	-1,518,790	-2,010,409	-1,879,896
Gap	0.78%	0.00%	0.24%	0.00%	1.28%	4.39%	6.78%
CPU	30	8	89	21	59	285	352
Spilled Pass.	824	1,008	1,735	713	1,223	1,943	1,308
Num. used AC	23	23	23	22	29	48	39
Best feas.							
B&B node	18	1	13	9	3	7	5
Cut iter.	1	0	8	3	3	2	4
Cuts inser.	1	0	15	11	5	5	12
Obj.	-1,180,671	-1,129,078	-1,193,168	-1,314,247	-1,518,790	-2,080,883	-1,959,598
Gap	0.63%	0.00%	0.00%	0.00%	1.28%	1.03%	2.83%
CPU	57	8	480	107	132	1,156	807
Spilled Pass.	797	1,008	1,735	713	1,223	1,260	1,092
Num. used AC	23	23	23	22	29	51	42

Table 10: Results for "Summer 1998" (fixed distribution of aircraft, with via flights)

because in general a via flight including two OD-pairs with low demands is cheaper than two non-via flights including one OD-pair each.

6.2.4 Results obtained by MINTO and CPLEX

We compared the lower bounds and feasible solutions obtained by our `BnP`-Optimizer with the results by the MINTO mixed-integer solver (Savelsbergh and Nemhauser [25], Version 3.0a, which is calling CPLEX [1] as a subroutine for solving the LP relaxations) and the CPLEX mixed-integer package, Version 6.0 (see [1]).

The computational results shown in the previous sections have been obtained by solving an aggregated (mixed-integer) version working on subfleet groups rather than solving the single (mixed-binary) aircraft formulation. I.e., if there are two aircraft of the same subfleet group serving the same rotation, the corresponding variable will take the value two in contrast to two single aircraft rotation variables taking the value one. Instead of working on individual aircraft for this comparison we only consider the case where an aircraft rotation is allowed to be flown at most once for each subfleet group, which is a restriction but yields a mixed-binary problem.

MINTO improved the lower bound in average by $\sim 1\%$ for the binary problem and $\sim 0.02\%$ for the integer version of the problem. The average gap MINTO sees is 8.53% for the binary problem, and 34.72% for the integer problem. No solutions for data sets "S98-Sa" and S98-Su" were found at all. CPLEX did not succeed in improving the bounds at all. The average gaps amount to 21.11% and 20.62%. (See Appendix for details.) In contrast to that, the `BnP` optimizer improved the lower bounds by 6.11% and 6.12% for the binary and integer problems, respectively, while average gaps amount to 0.93% and 0.80%.

7 Extensions

The basic assumption of our model is the even distribution of demands over the planning period, which is half a year in our case. Usually, in charter business the flights or complete vacation packages are booked for one up to several *weeks*. Assumption **A2** stems from this fact and leads to Assumptions **A3** and **A1**. Weekly schedules repeated for a complete season are not sufficient to model big changes in the demand caused, for example, by the beginning and end of school vacations. Our model can be extended by dropping the basic assumption and incorporating more detailed demand information. In this case, demands are not symmetric anymore, i.e., Assumption **A2** is dropped. Changes to the model and algorithm are twofold. First, the fixed aircraft rotation pattern has to be relaxed to include a) more rotations but still respects the via flight constraint (per rotation), or b) all rotations which are operationally feasible. Second, the return flight of each passenger must be ensured. I.e., if a passenger booked a flight to a destination abroad with a return flight n weeks (or even days in the extended model) later, it must be ensured that she/he *will* be transported back home again that date. Clearly, this also requires to choose a sufficiently long planning horizon.

It would be interesting to analyze the effects of this less restricted model on a) the instances with symmetric demands used in the previous section, and b) including big demand changes.

8 Conclusions

In this paper, we study a specific airline schedule generation problem which has not been presented in the literature before. We present a network design model which models the airline's current practice as a special case. We give a path-based mixed-integer programming formulation and present a solution approach that allows us to solve real world instances with nonlinear costs on aircraft rotations and passenger itineraries. Empirically good results confirm the tight approximation of the polyhedron conv(SGP) (i. e., the convex hull of all feasible solutions to SGP) by local min-cover inequalities. We demonstrate the efficiency of our approach for real world instances of a major charter airline. Our implementation solves most of the instances with an integrality gap of a very few percent and running times in the order of minutes on a customary PC.

References

[1] Using the CPLEX Callable Library, Version 6.0. CPLEX Optimization, Inc., 1998.

[2] A. Balakrishnan, T.L. Magnanti, and R.T. Wong. A Decomposition Algorithm for Local Access Telecommunications Network Expansion Planning. *Operations Research*, 43:58–76, 1995.

[3] C. Barnhart, E.L. Johnson, G.L. Nemhauser, M.W.P. Savelsbergh, and P.H. Vance. Branch-and-Price: Column Generation for Solving Huge Integer Programs. *Operations Research*, 46:316–329, 1998.

[4] C. Barnhart and R.R. Schneur. Air Network Design for Express Shipment Service. *Operations Research*, 44(6):852–863, 1996.

[5] K. Büdenbender, T. Grünert, and H.J. Sebastian. A Tabu Search Algorithm for the Direct Flight Network Design Problem. *To appear in* Transportation Science, 1999.

[6] S.-G. Chang and B. Gavish. Lower Bounding Procedures for Multiperiod Telecommunications Network Expansion Problem. *Operations Research*, 43(1):43–57, 1995.

[7] T. Christof, A. Löbel, and M. Stoer. PORTA – POlyhedral Representation Transfomation Algorithm. `www.zib.de/Optimization/Software/Porta/`, 1998.

[8] T.G. Crainic and G. Laporte. Planning models for freight transportation. *European Journal of Operational Research*, 97:409–438, 1997.

[9] H. Crowder, E.L. Johnson, and M.W. Padberg. Solving Large Scale Zero-One Linear Programming Problems. *Operations Research*, 31:803–834, 1983.

[10] M.S. Daskin and N.D. Panayotopoulos. A Lagrangian Relaxation Approach to Assigning Aircraft to Routes in Hub and Spoke Networks. *Transportation Science*, 23:91–99, 1989.

[11] G. Desaulniers, J. Desrosiers, I. Ioachim, M.M. Solomon, F. Soumis, and D. Villeneuve. A Unified Framework for Deterministic Time Constraint Vehicle Routing and Crew Scheduling Problems. *Les Cahiers de GERAD*, G-94-46, 1997.

[12] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* Freeman, 1979.

[13] K. Holmberg and J. Hellstrand. Solving the Uncapacitated Network Design Problem by a Lagrangean Heuristic and Branch-and-Bound. In U. Zimmermann, U. Derigs, W. Gaul, R.H. Möhring, and K.P. Schuster, editors, *Operations Research Proceedings 1996, Selected Papers of the SOR 96.* Springer, 1997.

[14] K. Holmberg and D. Yuan. A Lagrangean Heuristic Based Branch-and-Bound Approach for the Capacitated Network Design Problem. *Operations Research*, 46:247–259, 1998.

[15] B.W. Lamar, Y. Sheffi, and W.B. Powell. A Capacity Improvement Lower Bound for Fixed Charge Network Design Problem. *Operations Research*, 38(4):704–710, 1990.

[16] L.S. Lasdon. *Optimization Theory for Large Systems.* Collier-MaxMillan, New-York, 1970.

[17] T.L. Magnanti, P. Mirchandani, and R. Vachani. Modeling and Solving the Two-Facility Capacitated Network Loading Problem. *Operations Research*, 43:142–157, 1995.

[18] T.L. Magnanti, P. Mireault, and R.T. Wong. Tailoring Bender's decomposition for network design. *Mathematical Programming Study*, 26:112–154, 1986.

[19] T.L. Magnanti and R.T. Wong. Network design and transportation planning: Models and algorithms. *Transportation Science*, 18:1–55, 1984.

[20] M. Minoux. Network Synthesis and Optimum Network Design Problems: Models, Solution Methods and Applications. *Network*, 19:313–360, 1989.

[21] A. Noltemeier. Modelle und Lösungsverfahren zur Generierung von Flugplänen. PhD thesis, ZAIK/ZPR, University of Cologne, Germany, 1999. In preparation.

[22] M.W. Padberg, T.J. van Roy, and L.A. Wolsey. Valid Linear Inequalities for Fixed Charge Problems. *Operations Research*, 33:842–861, 1985.

[23] B. Rexing, C. Barnhart, T. Kniker, A. Jarrah, and N. Krishnamoorthy. Airline Fleet Assignment with Time Windows. To appear in *Transportation Science*, 1999.

[24] D.M. Ryan and B.A. Foster. An Integer Programming Approach to Scheduling. In A. Wren (ed.), *Computer Scheduling of Public Transport Urban Passenger Vehicle and Crew Scheduling.* North-Holland, Amsterdam, 35-139, 1981.

[25] M.W.P. Savelsbergh and G.L. Nemhauser. *Functional description of MINTO, a Mixed INTeger Optimizer, Version 3.0.* Georgia Institute of Technology, 1998.

[26] V. Sridhar and J.S. Park. Benders-and-Cut Algorithm for Fixed-Charge Capacitated Network Design Problem. To appear in *European Journal of Operational Research*, 1999.

[27] T.J. van Roy and L.A. Wolsey. Valid Inequalities for Mixed 0–1 Programs. *Discrete Applied Mathematics*, 14:199–213, 1986.

[28] P.H. Vance, A. Atamtürk, C. Barnhart, E. Gelman, E.L. Johnson, A. Krishna, D. Mahidhara, G.L. Nemhauser, and R. Rebello. A Heuristic Branch-and-Price Approach for the Airline Crew Pairing Problem. from author, 1997.

Appendix

Name	S98-Mo	S98-Tu	S98-We	S98-Th	S98-Fr	S98-Sa	S98-Su
Relaxation of (1) – (10)							
Root node							
LB	-1,230,547	-1,186,034	-1,242,441	-1,381,218	-1,592,579	-2,178,288	-2,118,325
BnP-Optimizer							
Root node							
#min-cover	38	29	27	42	72	139	130
LB w/cuts	-1,146,974	-1,120,285	-1,165,879	-1,274,329	-1,505,859	-2,057,953	-2,003,195
Bound impr.	6.79%	5.54%	6.31%	7.74%	5.44 %	5.52%	5.43%
Best feas.							
+#min-cover	0	2	9	3	2	4	0
Obj.	-1,134,839	-1,119,281	-1,163,921	-1,273,197	-1,484,201	-2,033,522	-1,953,032
CPU	35	8	76	46	31	704	438
Gap	1.06%	0.09%	0.17%	0.09%	1.44%	1.19%	2.50%
MINTO, Version 3.0a							
Root node							
#gen.-fc	84	61	89	74	112	191	152
LB w/cuts	-1,212,309	-1,173,888	-1,237,928	-1,360,490	-1,576,406	-2,160,934	-2,099,787
Bound impr.	1.48%	1.02%	0.36%	1.50%	1.02%	0.80%	0.88%
Cmp. to BnP	-5.70%	-4.78%	-6.18%	-6.76%	-4.68%	-4.77%	-4.60%
Best feas.							
#gen.-fc	769	781	969	854	1,128	919	907
Obj.	-1,135,283	-1,109,730	-1,126,033	-1,255,086	-1,434,582	-1,922,616	-1,896,551
B&B node	1,947	21,150	12,150	31,950	21,150	300	7,800
#nodes	131,206	114,691	17,434	46,319	24,437	2,501	8,612
CPU	175	1,505	5,819	4,997	6,534	1,577	7,015
Gap	6.35%	5.47%	9.04%	7.75%	9.00%	12.40%	9.68%
Cmp. to BnP	0.04%	-0.94%	-3.26%	-1.42%	-3.34%	-5.77%	-2.98%
CPLEX (MIP), Version 6.0							
Best feas.							
Obj.	-1,015,095	-1,046,896	-892,065	-1,193,269	-1,262,574	-1,468,056	-1,567,351
B&B node	151,680	137,370	48,550	1	95,350	1,370	8,520
#nodes	349,751	393,441	66,092	135,291	102,245	8,078	19,670
CPU	1,791	1,373	2,926	17	3,730	691	1,768
Gap	17.51%	11.73%	28.20%	13.61%	20.72%	32.61%	23.38%
Cmp. to BnP	-10.55%	-6.47%	-23.36%	-6.28%	-14.93%	-27.81%	-19.75%

Table 11: Comparison of BnP-Optimizer with MINTO and CPLEX, Part I
(Results for "Summer 1998" with fixed distribution of aircraft, w/o via flights, binary version);
(*#gen.-fc.*: number of generalized flow cover cuts inserted)

Name	S98-Mo	S98-Tu	S98-We	S98-Th	S98-Fr	S98-Sa	S98-Su
Relaxation of (1) – (10)							
Root node (int.)							
LB	-1,230,556	-1,186,034	-1,242,441	-1,381,389	-1,592,853	-2,178,304	-2,118,376
BnP-Optimizer							
Root node (int.)							
#min-cover	38	29	27	45	75	144	130
LB w/cuts	-1,146,974	-1.120.285	-1,165,879	-1,274,329	-1,505,859	-2,057,952	-2,003,195
Bound impr.	6.79%	5.54%	6.31%	7.75%	5.46%	5.53%	5.43%
Best feas.							
+#min-cover	3	2	3	0	1	4	0
Obj.	-1,136,116	-1,119,281	-1,163,921	-1,273,197	-1,486,976	-2,051,895	-1,947,594
CPU	123	11	59	34	407	487	2,168
Gap	0.95%	0.09%	0.17%	0.09%	1.25%	0.29%	2.78%
MINTO, Version 3.0a							
Root node							
#gen.-fc	3	2	2	0	2	1	1
LB w/cuts	-1,229,661	-1,185,565	-1,241,854	-1,381,389	-1,592,134	-2,178,275	-2,118,327
Bound impr.	0.07%	0.04%	0.04%	0%	0.05%	$\sim 0\%$	$\sim 0\%$
Cmp. to BnP	-7.21%	-5.92%	-6.52%	-8.40%	-5.73%	-5.85%	-5.75%
Best feas.							
#gen.-fc	106	86	17	45	37		
Obj.	-1,009,030	-1,068,683	732,477	731,477	679,769	n/a	n/a
B&B node	75,100	100,300	14,300	90,900	7,200		
#nodes	128,792	135,996	48,208	95,528	71,368		
CPU	4,317	4,968	2,608	7,422	841		
Gap	17.94%	9.86%	41.02%	47.05%	57.30%		
Cmp. to BnP	-11.19%	-4.52%	-37.07%	-42.55%	-54.29%		
CPLEX (MIP), Version 6.0							
Best feas.							
Obj.	1,043,790	-1,075,741	-936,315	-1,114,229	-1,293,978	-1,520,586	-1,548,376
B&B node	61,320	8,050	81,180	76,630	24,430	5,510	6,470
#nodes	402,541	392,793	82,600	119,394	98,211	8,670	19,110
CPU	638	89	3,932	2,554	1,001	2,535	1,320
Gap	15.18%	9.30%	24.64%	19.34%	18.76%	30.19%	26.91%
Cmp. to BnP	-8.13%	-3.89%	-19.56%	-12.49%	-12.98%	-25.89%	-20.50%

Table 12: Comparison of BnP-Optimizer with MINTO and CPLEX, Part II
(Results for "Summer 1998" with fixed distribution of aircraft, w/o via flights, integer version)