

Nissen, Rüdiger; Haase, Knut

Working Paper — Digitized Version

Duty-period-based network model for airline crew rescheduling

Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel, No. 581

Provided in Cooperation with:

Christian-Albrechts-University of Kiel, Institute of Business Administration

Suggested Citation: Nissen, Rüdiger; Haase, Knut (2004) : Duty-period-based network model for airline crew rescheduling, Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel, No. 581, Universität Kiel, Institut für Betriebswirtschaftslehre, Kiel

This Version is available at:

<https://hdl.handle.net/10419/147641>

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.

Manuskripte
aus den
Instituten für Betriebswirtschaftslehre
der Universität Kiel

Nr. 581
Duty-Period-Based Network Model for
Airline Crew Rescheduling

Rüdiger Nissen, Knut Haase

Februar 2004

Rüdiger Nissen, Christian-Albrechts-Universität zu Kiel, Institut für Betriebswirtschaftslehre,
Olshausenstr. 40, 24118 Kiel, Germany, nissen@bwl.uni-kiel.de
<http://www.bwl.uni-kiel.de/bwlinstitute/Prod/index.html>

Knut Haase, Technische Universität Dresden, Institut für Wirtschaft und Verkehr,
Mommsenstr. 13, 01062 Dresden, Germany, knut.haase@mailbox.tu-dresden.de
http://www.tu-dresden.de/vkiwv/vwibwl/bwl_hp.htm

Abstract

Airline rescheduling is a relatively new field in airline Operations Research but increasing amounts of traffic will make disturbances to the original schedule more frequent and more severe, and thus the need to address the various problems arising from this situation with systematic, cost-efficient approaches is becoming more urgent.

One such problem is crew rescheduling where after a disturbance in the crew schedule the aim is to determine new crew assignments that minimize the ‘impact’ on the original schedule. ‘Impact’ is most often defined as the number of changes in the crew schedule, but could also be seen as either the costs incurred by the changes or the period of time until full recovery is achieved.

In this work we present a new duty-period-based formulation for the airline crew rescheduling problem that uses a new type of resource constraints to efficiently cover the various labor regulations and that is tailored to the needs of European airlines, as well as a solution method based on branch-and-price. The solution method is tested on various rescheduling scenarios, each with several distinct cases. Especially solution time is a critical factor and the results show that the solution method is capable of providing solutions within the short period of time available to a rescheduler after a disturbance occurs.

Keywords: transportation, crew scheduling, rescheduling, irregular operations, disruption management, crew recovery, operational crew scheduling, column generation

1 Introduction

Many factors influencing the operation of an airline’s schedule have the potential to create disruptions (e.g., severe weather, aircraft mechanical problems, air traffic control system decisions). The most frequent cause for disruptions by far is inclement weather, which typically accounts for more than 70% of schedule deviations. These are also usually the most severe disruptions since the weather affects any flight entering or leaving the respective area.

The situation is exacerbated by the fact that overly optimized schedules offer so little slack that already minor disturbances can significantly affect the flight schedule, worsening the resulting delays and at the same time wiping out much of the cost reductions achieved through schedule optimization. Indeed, while excess crew cost has been driven down to almost zero, actual excess cost remains about 4-6%.

Recently research efforts have therefore been increasingly directed at developing optimization methods for airline rescheduling which is still mostly done manually by experienced schedulers. Of the different problems relevant for airline rescheduling the most challenging problem is probably crew rescheduling where after a disturbance in the crew schedule the aim is to determine new crew assignments that minimize the ‘impact’ on the original schedule. ‘Impact’ is most often defined as the number of changes in the crew schedule, but could also be seen as either the costs incurred by the changes or the period of time until full recovery is achieved.

On the one hand, crew rescheduling is challenging due to the difficulties in solving it, which arise from the complex regulations and problem sizes that also make crew scheduling so

interesting. On the other hand, finding good solutions to crew schedule disturbances is important due to the ‘human’ factor involved: frequent and wide-ranging changes to the original crew schedule would affect employee satisfaction in a negative way.

This paper presents a new model and solution method for crew rescheduling that is specifically tailored to the needs of European airlines with their distinct payment structure – as opposed to the more frequently treated U.S. system. The model uses a new type of resource constraints to efficiently cover the various labor regulations.

The paper is structured as follows: We first give a brief survey of the airline crew scheduling problem. We continue with an introduction to airline rescheduling in general, and crew rescheduling in particular. We then present a new crew rescheduling approach based on duty periods together with a solution method for this model based on the column generation principle. This is followed by the results from a series of computational experiments. We close with a summary.

2 Airline Crew Scheduling

The airline crew scheduling problem can be formulated as follows: Given an airline’s flight schedule, assign to each flight the necessary crew members for cockpit and cabin, so that the airline can operate its flight schedule at minimal cost for crews. The crew assignments have to take into consideration all relevant restrictions regarding working hours that are dictated mostly by government regulations, but also union agreements and the airline’s policies. Furthermore, capacity restrictions for the availability of crews have to be accounted for.

The way in which the cost of a crew schedule is determined depends mostly on the payment system for the crew members’ salaries. Whereas some airlines, especially European ones, rely on a system of fixed crew salaries, others, notably North American airlines, use a system called ‘pay-and-credit’ where the salary of a crew member is determined by his actual duty and flight time.

Even though the crew scheduling problem can easily be stated verbally, it is difficult to model and the sheer size of the problem adds to its difficulty. Thus, the problem is typically solved separately for cockpit and cabin crew, and then further divided into the two steps *crew pairing* and *crew assignment* which are solved sequentially.

Before we give a description of these two scheduling steps, we will define some terms that are used throughout this paper: A *flight leg* is a non-stop flight. A *duty period* is the period of time between reporting for an assignment and being released from that assignment. It is preceded and followed by a rest period. Thus, a duty period can basically be viewed as one workday. Flight legs are usually grouped into duty periods. A *pairing* is a sequence of flight legs that starts at a crew base and ends at the same crew base consisting usually of several duty periods. It is preceded and followed by a rest period. The duty periods in a pairing are separated by overnight stops. A pairing for short- and medium-haul problems is typically up to 5 days long whereas long-haul pairings may last longer. A situation when crew members travel as passengers on a flight leg to position themselves for their next flight or to head home at the end of a duty period is called *deadheading*.

2.1 Crew Pairing

Crew pairing is the first problem to be solved in crew scheduling. As was already mentioned, crew pairing is done separately for cockpit and cabin crew. Since pilots are usually only certified to fly one type of aircraft, it is possible to further reduce problem size by solving a separate subproblem for each of the different aircraft fleets the airline operates. For the cabin crew this is not always possible, because flight attendants are usually not bound to particular aircraft types. It is important to note that the crew pairing problem is solved on the level of a flight leg's entire crew and not on the individual crew-member level.

The aim of the crew pairing problem is to find a minimum cost set of pairings that cover all flights for the scheduling period (usually one month). Each pairing must satisfy all relevant government regulations, union agreements and other restrictions regarding duty time, flight time, and rest requirements.

The multitude of regulations and restrictions that have to be considered when checking a pairing for legality, and the complicated formulas for determining a pairing's cost are basically impossible to implement in an optimization model. And even if this were successful, the resulting model would become intractable. Thus, most solution approaches separate the process of generating the pairings from the selection of the least-cost subset (see. e.g., Anbil et al., 1991; Graves et al., 1993), and model the selection process as a set partitioning problem (SPP) or a set covering problem (SCP) (e.g., Hoffman, Padberg, 1993; Bixby et al., 1992). Only few approaches use heuristics to construct low-cost schedules (see. e.g., Wedelin, 1995). The most popular solution method in recent years has become the column generation approach (e.g., Lavoie et al., 1988; Vance et al., 1997; Desaulniers et al., 1997).

2.2 Crew Assignment

Crew assignment is the second problem to be solved in crew scheduling. Just as the crew pairing problem, the problem is usually decomposed by crew type (cockpit, cabin) and fleet. In some cases, it is further partitioned by crew bases. However, in contrast to the pairing problem, the crew assignment problem is solved on the individual crew-member level.

The crew assignment problem can be stated as follows: Given a set of pairings to cover the airline's flights for the planning period (usually about one month), assign individual crew members (for cockpit and cabin) to each pairing so that all on-board functions are adequately staffed. The crew assignments have to consider preassigned activities for each individual, such as training periods, holidays, medical exams, etc., as well as to comply with the relevant set of rules resulting from government regulations and union agreements.

Different airlines pursue different aims for the crew assignment problem, but in general they constitute a combination of cost minimization and maximization of 'quality of life' criteria. 'Quality of life' is then calculated as the negative cost trade-off the airline is willing to pay for crew satisfaction.

Depending on how and when airlines consider crew requests or preferences in their scheduling process, one can distinguish three general methodologies, *bidline generation*, *rostering*, and *preferential bidding*.

The solution approaches are usually similar for all three methodologies and the SPP can generally be applied to all three as well. Solution methods for bidline generation are presented in, e.g., Jarrah, Diamond (1997), Campbell et al. (1997); rostering is treated by, e.g., Day, Ryan (1997), Gamache et al. (1999); and preferential bidding has been studied in Gamache et al. (1998).

3 Airline Rescheduling

As soon as an airline starts to carry out its flight schedule, various unexpected events occur and lead to disruptions of the schedule. Due to the near optimal schedules produced by the initial scheduling process under the assumption of a deterministic situation, only very little slack is left to allow for these disruptions to be easily absorbed. Furthermore, the resources in the flight and crew schedules are tightly linked together, so that perturbations most commonly affect both schedules. This leads to the danger of system nervousness where even small deviations can cascade through the airline's network, leading to even more deviations, in the process erasing most of the cost savings realized by the initial scheduling and impeding the airline's ability to meet its commitments.

Many factors influencing the operation of an airline's schedule have the potential to create disruptions (e.g., severe weather, aircraft mechanical problems, air traffic control system decisions). The most frequent cause for disruptions by far is inclement weather, which typically accounts for more than 70% of schedule deviations. These are also usually the most severe disruptions since the weather affects any flight entering or leaving the respective area. Situations where a disruption is significant enough to result in rescheduling are called *irregular operations*. The airline's Operations Control Center (OCC) is in charge of handling all such disruptions.

During irregular operations most airlines will try to be back on the original flight schedule as fast as possible, since a high level of on-time performance is important for the airline's customer reputation and thus directly affects the airline's market share.

The OCC can react to irregular operations in the flight schedule by adjusting flight speeds, by shortening aircraft ground turn times, by swapping aircraft between flights, by using spare aircraft or by canceling or delaying flights. Solution methods for rescheduling of the flight schedule are discussed in, e.g., Yan, Lin (1997), Bard et al. (2001), and Rosenberger (2003).

Since the crew schedule is not published to the general public, the foremost concern of the OCC here is not a fast, but a low cost, least disturbance recovery in case of deviations. In some cases, however, an airline also may place a high priority on a fast recovery of the crew schedule to prevent disturbances from propagating too much into the future (Irrgang 1995).

Crew rescheduling is done after the flight schedule has been fixed. The OCC can react by extending crew working hours, by swapping crews between flights or by calling in reserve

crews. For any change in the crew schedule, it has to be assured that the new crew duties are still permissible with regard to government regulations and union agreements. In some cases, when it is not possible to find a legal crew assignment, the OCC has to go back and find other scheduling alternatives for the flight schedule.

4 Crew Rescheduling

In contrast to crew scheduling, crew rescheduling (also known as *operational crew scheduling* or *crew recovery*) has so far been treated only on a few occasions, despite the fact that irregular operations may drastically change the original schedule and despite the sensitivity of crew members to system nervousness. As a result, it is still mostly done manually by experienced schedulers, with only minor support from the sophisticated tools involved in the initial scheduling process.

The crew rescheduling problem can be formulated as follows: After a disturbance in the crew schedule has occurred, choose a set of flights that can have their crew assignments changed. Determine new crew assignments for these flights that minimize the ‘impact’ on the original schedule. ‘Impact’ is most often defined as the number of changes in the crew schedule, but could also be seen as either the costs incurred by the changes or the period of time until full recovery is achieved. It is usually assumed that a feasible flight schedule is available as input data, i.e., the flight rescheduling problem has been solved first.

Stojković, Soumis (1998) present an approach that aims to generate as quickly as possible personalized crew pairings by solving the crew pairing and the crew assignment problem simultaneously. The objective is to cover all flights (and other tasks) at minimal cost while retaining as much as possible of the original crew schedule. They solve the problem of assigning flights to crew candidates with a column generation approach where the master problem is a modified SPP that allows over- and (in the case of cabin crew) undercovering of flights and has additional restrictions for global constraints. The subproblems are resource-constrained shortest path problems where a separate duty-period-based graph is constructed for each crew candidate with duty periods represented as nodes. The authors report computational results for problems with up to 16 crew candidates and 210 tasks of which up to 114 were not frozen. Solution times ranged from a few seconds to 20 minutes.

Wei et al. (1997) introduce a heuristic to find as fast as possible a solution that covers all flights with the objective of changing as few pairings as possible. The method, which works on the crew level rather than the individual crew-member level, is a depth-first branch-and-bound search procedure. At each node in the search tree, a set of uncovered flights and a list of modified pairings represents the current problem state. As long as there are uncovered flights, one of the uncovered flights is selected according to heuristic rules and a candidate crew list is created to cover this flight. For each flight/crew assignment, a new branch in the search tree is created. Since it is not allowed to have broken pairings at any time during the search process, a shortest-path procedure that favors retaining as much of the old pairing as possible is used to create a new pairing (in case of reserve crews) or repair the crew’s current pairing after it has been assigned an uncovered flight. Since this may require the crew to skip some of the previously assigned flights, new flights

may be added to the set of uncovered flights. The search procedure continues until either a time limit has been exceeded or a predetermined number of solutions was found. The number of modified pairings at a node is used to prune the search tree. If no full solution could be found (i.e., not all flights could be covered), it is also possible to output partial solutions. Limited computational results are presented for a set of test problems with up to 51 flights over a two-day period, which could be solved within a few seconds. Yu et al. (2003) present a real-time DSS called CrewSolver based on a refined version of this solution method. They give computational results for problems with up to 40 affected flights on a single day which could be solved within a few minutes. They also report about the successful use of the system at Continental Airlines in various real-life situations, e.g. the recovery from the September 11th terrorist attacks.

Lettovský et al. (2000) describe a crew-level optimization approach that finds a minimum cost reassignment of crews to a disrupted flight schedule. First, a combination of a predefined time window and a maximum number of candidate crews per misconnection is used to limit the search for a good set of crews to cover all flights. Efficient heuristics are used to select possible deadhead flights without which it frequently would be impossible to find a recovery solution. To choose a minimum cost set of pairings, they then use a modified SCP that includes additional variables for flight cancellations and deadheads which is solved by a column generation scheme. Pairings are generated individually for each crew as extensions of the flown part of the crew's original pairing. Pairing costs are based on the difference of the new and the original pairing's costs, with a bonus subtracted for each flight that is reassigned to the same crew as before the disruption. Integer solutions are obtained in a branch-and-bound framework. Computational results are given for a case study with three different scenarios based on schedule data from a major U.S. carrier. The problems involved up to 38 crews and 122 flight legs and were all solved in less than 2 minutes.

5 Duty-Period-Based Network Model

The most important distinguishing factor between crew scheduling approaches in Europe and the U.S. is the airlines' payment system. Since our objective is to develop a crew rescheduling approach for European airlines where fixed crew salaries are predominant, we do not have to follow the concept of previous approaches and attempt to repair the several-day-long pairings. Instead we will work with one-day duty periods. This allows us to use shorter rescheduling horizons which translate into smaller problems and thus faster solution times.

Since the model does not use pairings, it can also integrate the two steps of crew pairing and assignment from the initial scheduling process into one. This has the additional advantage that the model can be used for rescheduling on the level of entire crews as well as the level of individual crew members. The former is preferable if the crew shall remain unchanged by the rescheduling process, whereas the latter is needed in case of disturbances emanating from the crews themselves, e.g., if a crew member falls sick.

5.1 An Example for Rescheduling with Duty Periods

Assume a small airline catering to business travelers that employs seven crews, five of them based in Hamburg and two in Munich. Each crew consists of a pilot, a copilot, and three flight attendants. The airline operates the flights as given in Table 1 from Monday through Friday, and does not fly on the weekend.

Table 1: One-Day Flight Schedule

Flight	Departure	Arrival
f_1	HAM 06:00	FRA 08:00
f_2	FRA 09:00	MUC 10:00
f_3	MUC 12:00	HAM 14:00
f_4	HAM 08:00	FRA 10:00
f_5	FRA 12:00	HAM 14:00
f_6	MUC 08:00	FRA 09:00
f_7	FRA 11:00	MUC 12:00
f_8	HAM 15:00	FRA 17:00
f_9	FRA 18:00	MUC 19:00
f_{10}	MUC 20:00	HAM 22:00
f_{11}	HAM 16:00	FRA 18:00
f_{12}	FRA 20:00	HAM 22:00
f_{13}	MUC 16:00	FRA 17:00
f_{14}	FRA 19:00	MUC 20:00

In scheduling their crews, the airline has to adhere to the following regulatory framework:

- A crew member's duty period may last up to 14 hours.
- After each duty period, a rest period of at least 10 hours is mandatory.
- Within each 7-day period a rest period of at least 36 hours has to be granted. (This rule is automatically adhered to by not flying on the weekend.)
- Before the first flight in a duty, one hour has to be scheduled for briefing the crew.
- After the last flight in a duty, one hour has to be scheduled for debriefing the crew.
- Deadheading is considered duty time.

In addition, the schedule has to observe 30 minutes of minimum transfer time between two consecutive flights in a duty period to allow a crew to prepare its next flight and change gates, if necessary.

Let us assume the airline has generated a one-week solution for this crew scheduling problem as given in Table 2.

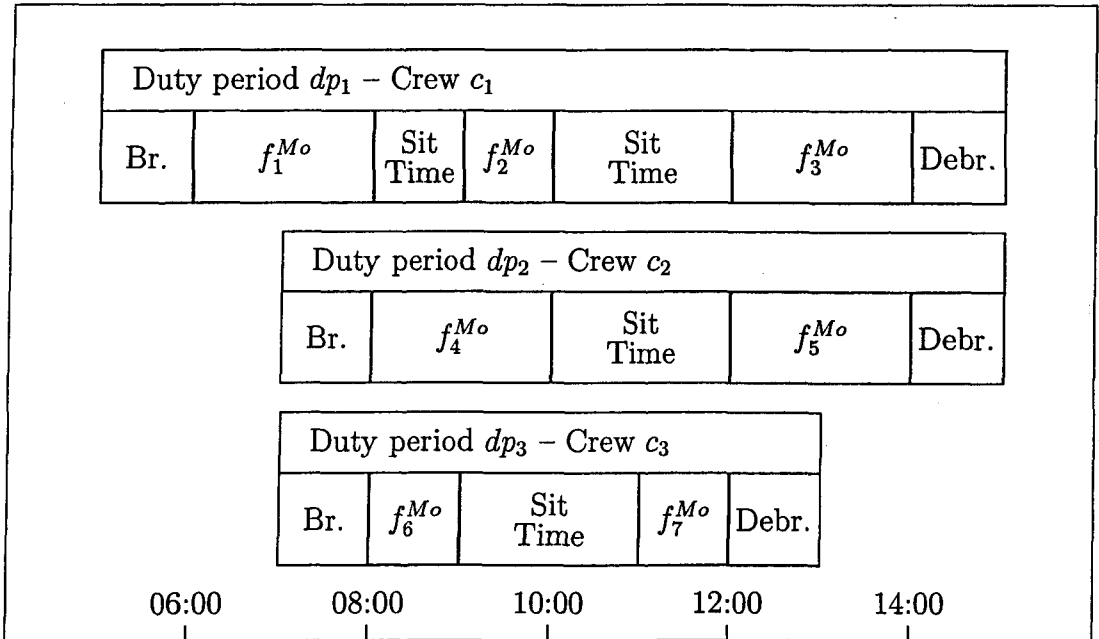
From a duty period point of view this means that each day six different regular duty periods are operated, whereas the reserve crew will only have a duty period assigned to

Table 2: One-Week Crew Schedule

Crew	Flights Assigned				
	Mo	Tu	We	Th	Fr
c_1	$f_1^{Mo}, f_2^{Mo}, f_3^{Mo}$	f_4^{Tu}, f_5^{Tu}	$f_1^{We}, f_2^{We}, f_3^{We}$	f_4^{Th}, f_5^{Th}	$f_1^{Fr}, f_2^{Fr}, f_3^{Fr}$
c_2	f_4^{Mo}, f_5^{Mo}	$f_1^{Tu}, f_2^{Tu}, f_3^{Tu}$	f_4^{We}, f_5^{We}	$f_1^{Th}, f_2^{Th}, f_3^{Th}$	f_4^{Fr}, f_5^{Fr}
c_3	f_6^{Mo}, f_7^{Mo}	f_6^{Tu}, f_7^{Tu}	f_6^{We}, f_7^{We}	f_6^{Th}, f_7^{Th}	f_6^{Fr}, f_7^{Fr}
c_4	$f_8^{Mo}, f_9^{Mo}, f_{10}^{Mo}$	f_{11}^{Tu}, f_{12}^{Tu}	$f_8^{We}, f_9^{We}, f_{10}^{We}$	f_{11}^{Th}, f_{12}^{Th}	$f_8^{Fr}, f_9^{Fr}, f_{10}^{Fr}$
c_5	f_{11}^{Mo}, f_{12}^{Mo}	$f_8^{Tu}, f_9^{Tu}, f_{10}^{Tu}$	f_{11}^{We}, f_{12}^{We}	$f_8^{Th}, f_9^{Th}, f_{10}^{Th}$	f_{11}^{Fr}, f_{12}^{Fr}
c_6	f_{13}^{Mo}, f_{14}^{Mo}	f_{13}^{Tu}, f_{14}^{Tu}	f_{13}^{We}, f_{14}^{We}	f_{13}^{Th}, f_{14}^{Th}	f_{13}^{Fr}, f_{14}^{Fr}
c_7	Reserve	Reserve	Reserve	Reserve	Reserve

it when it is called upon in the case of irregular operations. To illustrate the composition of the duty periods, those flown by crews c_1 to c_3 on Monday are shown in Figure 1. For example, duty period dp_1 starts in Hamburg at 05:00 with a one hour briefing. It then covers flight f_1^{Mo} to Frankfurt where the crew has one hour of sit time before continuing on flight f_2^{Mo} to Munich. There they have to wait for another hour before they serve flight f_3^{Mo} back to Hamburg. In Hamburg the duty period finishes with a one hour debriefing, so that the crew's workday ends at 15:00.

Figure 1: Composition of Duty Periods for Crews c_1 to c_3 on Monday



Let us now assume that on Wednesday flight f_2^{We} cannot take off as scheduled due to a technical problem, but instead will leave with a 2-hour delay, thus not arriving in Munich in time for the crew to catch its next scheduled flight, i.e., flight f_3^{We} . The airline's Operations Control Center does not want that flight to be delayed as well, so it has to

try and find a new crew schedule that allows all other remaining flights to take place as published in the airline's flight schedule. This new schedule should deviate from the old schedule as little as possible. Since crews c_1 to c_3 operate all the morning flights, whereas crews c_4 to c_6 operate all the afternoon flights, the OCC decides that they will try to find a solution that does not affect the afternoon flights at all and that brings all crews back to their bases at the end of their Friday duty period. It also wants to keep the crews together, which means that rescheduling must be done on the crew level.

Table 3: Example: Duty Periods Generated for Rescheduling

Wednesday	Thursday	Friday
$(dp_1 : f_1^{We} / c_1)$	$dp_{17} : f_1^{Th}$	$(dp_{33} : f_1^{Fr})$
$dp_2 : f_1^{We}, f_2^{We} \text{ (delayed)} / c_1$	$dp_{18} : f_1^{Th}, f_2^{Th}$	$dp_{34} : f_1^{Fr}, f_2^{Fr}$
	$dp_{19} : f_1^{Th}, f_2^{Th}, f_3^{Th}$	$dp_{35} : f_1^{Fr}, f_2^{Fr}, f_3^{Fr}$
$(dp_3 : f_1^{We}, f_5^{We} / c_1)$	$dp_{20} : f_1^{Th}, f_5^{Th}$	$dp_{36} : f_1^{Fr}, f_5^{Fr}$
$(dp_4 : f_1^{We}, f_7^{We} / c_1)$	$dp_{21} : f_1^{Th}, f_7^{Th}$	$dp_{37} : f_1^{Fr}, f_7^{Fr}$
$(dp_5 : f_2^{We} \text{ (delayed)})$	$dp_{22} : f_2^{Th}$	$dp_{38} : f_2^{Fr}$
	$dp_{23} : f_2^{Th}, f_3^{Th}$	$dp_{39} : f_2^{Fr}, f_3^{Fr}$
$dp_6 : f_3^{We}$	$dp_{24} : f_3^{Th}$	$dp_{40} : f_3^{Fr}$
$dp_7 : f_4^{We} / c_2$	$dp_{25} : f_4^{Th}$	$(dp_{41} : f_4^{Fr})$
$(dp_8 : f_4^{We}, f_2^{We} \text{ (delayed)} / c_1, c_2)$		
$dp_9 : f_4^{We}, f_5^{We} / c_2$	$dp_{26} : f_4^{Th}, f_5^{Th}$	$dp_{42} : f_4^{Fr}, f_5^{Fr}$
$dp_{10} : f_4^{We}, f_7^{We} / c_2$	$dp_{27} : f_4^{Th}, f_7^{Th}$	$dp_{43} : f_4^{Fr}, f_7^{Fr}$
$(dp_{11} : f_5^{We})$	$dp_{28} : f_5^{Th}$	$dp_{44} : f_5^{Fr}$
$dp_{12} : f_6^{We} / c_3$	$dp_{29} : f_6^{Th}$	$(dp_{45} : f_6^{Fr})$
$(dp_{13} : f_6^{We}, f_2^{We} \text{ (delayed)} / c_1, c_3)$		
$dp_{14} : f_6^{We}, f_5^{We} / c_3$	$dp_{30} : f_6^{Th}, f_5^{Th}$	$dp_{46} : f_6^{Fr}, f_5^{Fr}$
$dp_{15} : f_6^{We}, f_7^{We} / c_3$	$dp_{31} : f_6^{Th}, f_7^{Th}$	$dp_{47} : f_6^{Fr}, f_7^{Fr}$
$(dp_{16} : f_7^{We})$	$dp_{32} : f_7^{Th}$	$dp_{48} : f_7^{Fr}$

As was shown in Section 3, a typical OCC would find a solution to this problem manually, by looking at possible crew substitutions to cover flight f_3^{We} , and ways of fixing the schedule. Using the approach proposed in this Section, we generate all possible duty periods with the morning flights for the remaining days of the week. This yields the duty periods listed in Table 3.

Duty periods $dp_1 - dp_{16}$ cover Wednesday, $dp_{17} - dp_{32}$ Thursday, and $dp_{33} - dp_{48}$ Friday, respectively. For example, dp_1 , dp_{17} and dp_{33} are identical, except that they cover different days. Overall 48 duty periods can be generated.

However, some duty periods can be disregarded (shown in brackets in Table 3). For duty periods dp_5 , dp_{11} and dp_{16} this is the case, because no crew without prior engagement is located at the respective departure airports, so that these duty periods cannot be covered under any circumstances. And since all crews should return to their home bases by the end of Friday, all duty periods not ending at a crew base can be disregarded as well. This concerns the duty periods dp_{33} , dp_{41} and dp_{45} .

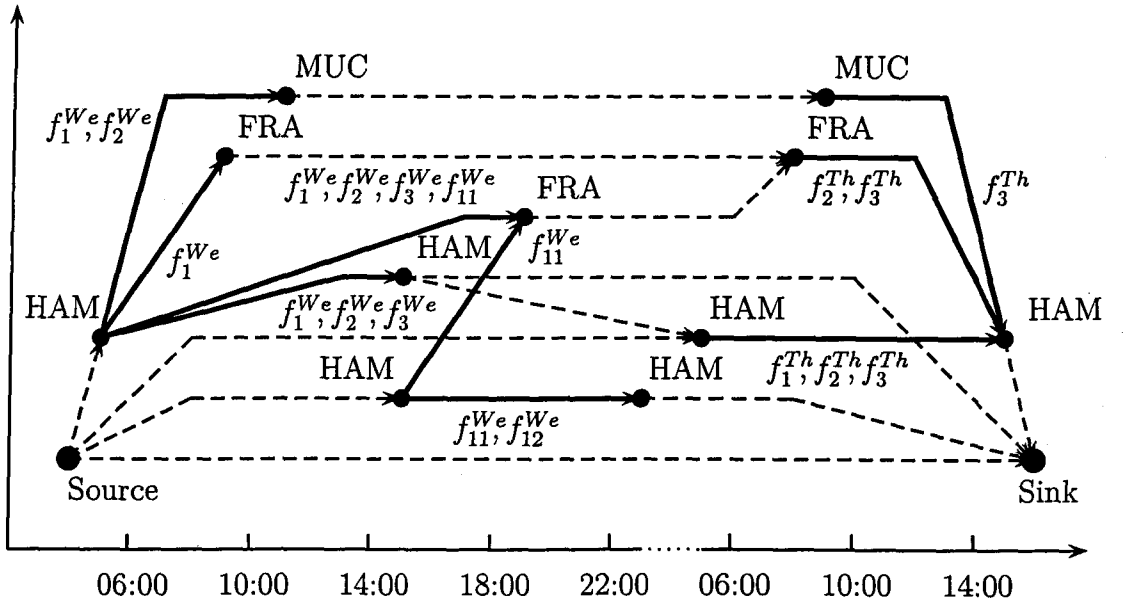
For some duty periods, we will not be able to freely choose a crew assignment, but instead have to fix the assignment a priori (indicated in Table 3 by noting the preassigned crew

together with the duty period). For example, when the OCC gets notice of the delay of flight f_2^{We} , crew c_1 has already completed flight f_1^{We} and begun the preparations for flight f_2^{We} . Thus, all duty periods containing these flights would have to be carried out by crew c_1 and any duty period for crew c_1 has to contain both flights. This concerns duty periods $dp_1 - dp_5$, dp_8 and dp_{13} , of which all but dp_2 can be disregarded because they do not cover both flights. Similarly, duty periods $dp_8 - dp_{10}$ would have to be serviced by crew c_2 , and duty periods $dp_{12} - dp_{15}$ would need to be operated by crew c_3 . As one can see this leads to the fact that duty periods dp_8 and dp_{13} also cannot be selected, because the first flights in these duty periods have already been carried out by crews c_2 and c_3 , respectively, whereas the last flight would have to be carried out by crew c_1 .

The next step after generating the duty periods is to generate a time-space network with them. Duty periods are represented as arcs. Thus, a node represents an airport at a specific time. Apart from the duty period arcs we also need rest arcs to get a connected graph. Finally, we need dummy nodes for each crew at the beginning and the end of the rescheduling period as artificial network source and sink, respectively. These are connected to the rest of the graph via dummy arcs.

The network for our example consists of 35 nodes and 94 arcs. Since this network is too large to show here, Figure 2 illustrates instead the structure of the network that would result for a crew located in Hamburg if only flights f_1^{We} , f_2^{We} , f_3^{We} , f_{11}^{We} , f_{12}^{We} on Wednesday and f_1^{Th} , f_2^{Th} , f_3^{Th} on Thursday were considered. Thick, solid lines indicate duty period arcs, whereas thin, dashed lines indicate rest period arcs and dummy arcs.

Figure 2: Network for Rescheduling



Using such a network, we can solve the rescheduling problem by finding a path for each crew from its source to its sink so that overall all flights are covered and the new schedule is as close as possible to the original crew schedule.

The best possible solution that uses only the morning flights would be to cancel flights f_3^{We} and f_4^{Th} . Crew c_1 would then stay in Munich after finishing flight f_2^{We} and on Thursday would deadhead on flight f_6^{Th} to Frankfurt to catch up with its old schedule by carrying out flight f_5^{Th} . Crews c_2 and c_3 would not be affected by this.

Since cancelling flights is not desirable, the OCC decides to try and find a better solution by also considering the afternoon flights. This means that now overall 248 duty periods can be generated, of which 55 can be disregarded. From the remaining 193 duty periods we can generate a network with 68 nodes and 376 arcs, and a new solution is found that allows to carry out all flights. The new schedule is shown in Table 4 with changes to the assignments of crews c_1 , c_3 and c_6 shown bold. Crew c_1 will debrief after flight f_2^{We} on Wednesday and then on Thursday it will deadhead on f_6^{Th} instead of operating f_4^{Th} , before it continues with its old schedule. Crew c_3 will, in addition to its normally scheduled flights, also serve flights f_{13}^{We} , f_{14}^{We} , f_{13}^{Th} and f_{14}^{Th} . Crew c_6 will fly only flight f_3^{We} on Wednesday, flight f_4^{Th} on Thursday and then on Friday deadhead on flight f_2^{Fr} to Munich to catch up with its old schedule. Overall 6 flights have to be reassigned to other crews compared to the original schedule and 2 deadhead flights are added to the schedule.

Table 4: Example: Rescheduled Crew Schedule

Crew	Flights Assigned		
	We	Th	Fr
c_1	f_1^{We}, f_2^{We}	f_6^{Th}, f_5^{Th}	$f_1^{Fr}, f_2^{Fr}, f_3^{Fr}$
c_2	f_4^{We}, f_5^{We}	$f_1^{Th}, f_2^{Th}, f_3^{Th}$	f_4^{Fr}, f_5^{Fr}
c_3	$f_6^{We}, f_7^{We}, f_{13}^{We}, f_{14}^{We}$	$f_6^{Th}, f_7^{Th}, f_{13}^{Th}, f_{14}^{Th}$	f_6^{Fr}, f_7^{Fr}
c_4	$f_8^{We}, f_9^{We}, f_{10}^{We}$	f_{11}^{Th}, f_{12}^{Th}	$f_8^{Fr}, f_9^{Fr}, f_{10}^{Fr}$
c_5	f_{11}^{We}, f_{12}^{We}	$f_8^{Th}, f_9^{Th}, f_{10}^{Th}$	f_{11}^{Fr}, f_{12}^{Fr}
c_6	f_3^{We}	f_4^{Th}	$f_2^{Fr}, f_{13}^{Fr}, f_{14}^{Fr}$

5.2 An Example for the Use of Resources

One of the most complicated issues in crew (re-)scheduling is to assure that a crew schedule is legal, i.e., that it complies with all work rules and preassignments.

In the previous example we mentioned that some duty periods could only be assigned to certain crews, e.g., because of already begun duties. However, we did not detail how this could be achieved.

Thus, in this section we will show how the concept of arc-subset-based resources can be used to enforce such preassignments. Nissen (2003) describes in detail how to use resources to achieve the even more important compliance with all work rules that cannot be covered otherwise in the duty periods or the network.

Arc-subset-based resources represent the transfer of partially renewable resources to networks. Partially renewable resources were conceived to overcome the deficiencies of existing resource concepts, i.e., renewable and non-renewable resources, for time-horizon-based scheduling (Böttcher et al., 1999). Whereas renewable resources are constrained on single

periods of the planning horizon, non-renewable resources are not constrained in every period but for the complete planning horizon. The weakness of these concepts becomes evident when complex rules and requirements are introduced, e.g., in the context of staff scheduling. For example the situation that an employee should have his lunch break in any one of several periods cannot be modeled, because the resource-availability needs to be constrained over a subset of periods containing more than one but not all elements. Partially renewable resources overcome this weakness by constraining the availability of a resource over any subset of periods. Renewable and non-renewable resources are then special cases of partially renewable resources.

When we want to restrict resource usage in a network rather than for a scheduling horizon, we do not face these modeling difficulties. Corresponding to renewable and non-renewable resources, known network resource types may restrict consumption either over a single arc or over the full arc set. By restricting resource usage over the full path we are able to model situations like the one described above or the work rules encountered in crew scheduling. However, by introducing arc-subset-based resources that limit capacities over a subset of arcs – rather than over subsets of periods as in partially renewable resources – we can develop more efficient algorithms by being able to determine the ‘final status’ of a resource’s consumption possibly before the end of the path.

Formally arc-subset-based resources in a network $G = (\mathcal{N}, \mathcal{A})$ can be defined as follows:

- A set of resources \mathcal{R} is given.
- For each resource $r \in \mathcal{R}$ we define the set of arc-subset indices Π_r and for each $\pi \in \Pi_r$ the arc-subset $\mathcal{A}_{r\pi} \subseteq \mathcal{A}$.
- For each resource $r \in \mathcal{R}$ the capacity that is available for crew $p \in \mathcal{P}$ over the arc-subset $\mathcal{A}_{r\pi} \subseteq \mathcal{A}$ with $\pi \in \Pi_r$ is limited to $K_{rp\pi}$ units. The usage of resource $r \in \mathcal{R}$ by arc $ij \in \mathcal{A}$ and crew member $p \in \mathcal{P}$ is k_{rpij} units.

Assuring that preassignments are adhered to is then done by creating a ‘preassignment’ resource with one arc-subset that contains all duty period arcs which have to be carried out by a certain crew member. In our example these are the arcs for duty periods dp_2 for crew c_1 , dp_7, dp_9 and dp_{10} for crew c_2 , and dp_{12}, dp_{14} and dp_{15} for crew c_3 . The available capacity for that resource is set to 0 and the resource usage of each arc in the subset is set to 0 for the crew that can carry the duty period out and to 1 for all others. For example, the resource consumption of the arc corresponding to duty period dp_2 is set to 0 for crew c_1 and to 1 for all other crews. This assures that the available capacity will be exceeded if this duty period is assigned to any other crew than c_1 .

5.3 Model Formulation

To formulate the duty-period-based network model for airline crew rescheduling, we make the following assumptions:

1. Rescheduling is done separately for each fleet.

2. The regulations restricting the scheduling of crews/crew members are known.
3. Deadheading is allowed.
4. A set \mathcal{F} of flights is given that have to be covered. The flight schedule contains for each flight the departure and arrival times and airports. Each flight has to be staffed with a crew qualified for flying the assigned aircraft. It is known which crew was scheduled to cover the flight according to the original crew schedule and whether that assignment can be changed.
5. A set of crews/crew members \mathcal{P} for whom at least some of their flight assignments can be rescheduled is given. For each crew/crew member it is known where it is located at the beginning of the rescheduling period, where it has to be at the end of that period and to which home base it is assigned. Their workload already finished prior to the rescheduling period is known as well.
6. From the set of flights \mathcal{F} a set of duty periods can be generated. Each duty period lasts approximately one workday and adheres to all relevant regulations.
7. A network $G = (\mathcal{N}, \mathcal{A})$ can be set up with a node set \mathcal{N} and an arc set \mathcal{A} . A node $i \in \mathcal{N}$ represents an airport at a specific time. In addition, there are artificial source and sink nodes so that each crew member $p \in \mathcal{P}$ can be assigned a source and a sink. An arc $ij \in \mathcal{A}$ is either: a duty period, a legal rest period separating two duty periods, an arc leading from a source node to a duty period or from a duty period to a sink node, or an arc connecting a source and a sink node. Two duty period arcs ij and kl are only connected by a rest period arc jk , if a crew is allowed to serve duty period kl after serving duty period ij .
8. Each node $i \in \mathcal{N}$ has a supply of s_{ip} for crew/crew member $p \in \mathcal{P}$: The crew's/crew member's source node has a supply $s_{ip} = 1$, whereas its sink node has demand of $s_{ip} = -1$. All others nodes are transitory nodes with $s_{ip} = 0$.
9. For each flight $f \in \mathcal{F}$ the subset of arcs $\mathcal{A}_f \subseteq \mathcal{A}$ contains those arcs representing a duty period covering that flight.
10. A set of resources \mathcal{R} is given.
11. For each resource $r \in \mathcal{R}$ we define the set of arc-subset indices Π_r and for each $\pi \in \Pi_r$ the arc-subset $\mathcal{A}_{r\pi} \subseteq \mathcal{A}$.
12. For each resource $r \in \mathcal{R}$ the capacity that is available for crew/crew member $p \in \mathcal{P}$ over the arc-subset $\mathcal{A}_{r\pi} \subseteq \mathcal{A}$ with $\pi \in \Pi_r$ is limited to $K_{rp\pi}$ units. The usage of resource $r \in \mathcal{R}$ by arc $ij \in \mathcal{A}$ and crew member $p \in \mathcal{P}$ is k_{rpij} units.
13. The costs of changing the original schedule have to be considered. Change costs are incurred by arcs, that is, the cost of arc $ij \in \mathcal{A}$ if assigned to crew member $p \in \mathcal{P}$ is c_{pij} .

Using these assumptions, a duty-period-based network model can be formulated for re-scheduling on the level of crews as well as crew members.

The decision variables of the model are:

$$x_{pij} = \begin{cases} 1 & \text{if arc } ij \in \mathcal{A} \text{ is assigned to crew } p \in \mathcal{P} \\ 0 & \text{otherwise} \end{cases}$$

With these variables and the parameters introduced in the assumptions, the model for crew-level rescheduling can be stated as follows:

$$\min \sum_{p \in \mathcal{P}} \sum_{ij \in \mathcal{A}} c_{pij} x_{pij} \quad (1)$$

subject to

$$\sum_{p \in \mathcal{P}} \sum_{ij \in \mathcal{A}_f} x_{pij} \geq 1 \quad (f \in \mathcal{F}) \quad (2)$$

$$\sum_{ij \in \mathcal{A}_{r\pi}} k_{rpij} x_{pij} \leq K_{r\pi} \quad (p \in \mathcal{P}; r \in \mathcal{R}; \pi \in \Pi_r) \quad (3)$$

$$\sum_{ij \in \mathcal{A}} x_{pij} - \sum_{ji \in \mathcal{A}} x_{pji} = s_{ip} \quad (p \in \mathcal{P}; i \in \mathcal{N}) \quad (4)$$

$$x_{pij} \in \{0, 1\} \quad (p \in \mathcal{P}; ij \in \mathcal{A}) \quad (5)$$

The objective function (1) minimizes the costs that are incurred by changing each crew's original schedule. Constraints (2) assure that each flight is staffed with a crew, and allow other crews to deadhead on that flight. (3) assure that resource limitations are taken into account for each crew in each arc-subset. For each resource $r \in \mathcal{R}$, crew $p \in \mathcal{P}$, and the respective arc-subset $\pi \in \Pi_r$, the capacity usage does not exceed the available capacity $K_{r\pi}$. (4) model the flow of crews through the network. (5) define the allowed values for the variables.

If the model is to be used on the crew-member level, the following additional assumptions have to be made:

14. A set of crew-member qualification groups \mathcal{Q} is given.
15. The subset of crew members $\mathcal{P}_q^Q \subseteq \mathcal{P}$ belonging to qualification group $q \in \mathcal{Q}$ is known.
16. Each flight $f \in \mathcal{F}$ has to be staffed with d_{fq} crew members from qualification group $q \in \mathcal{Q}$.

Constraints (2) then have to be replaced with constraints (6) which assure that each flight is staffed with enough crew members from all qualification groups, while again allowing other crew members to use that flight as a deadhead.

$$\sum_{p \in \mathcal{P}_q^Q} \sum_{ij \in \mathcal{A}_f} x_{pij} \geq d_{fq} \quad (f \in \mathcal{F}; q \in \mathcal{Q}) \quad (6)$$

Nissen (2003) shows that the duty-period-based network model for airline crew rescheduling (1) to (5) as well as its modification with (6) are NP-hard and that current labor regulations can be covered with resource restrictions (3).

Since problem sizes for this model are too large to be solved directly, a column generation approach was chosen instead. Applying Dantzig-Wolfe decomposition to the original model, we get a column generation formulation with the following master problem where each column represents a legal crew schedule:

$$\min \sum_{e \in \mathcal{E}} c_e y_e \quad (7)$$

$$\text{s.t.} \quad \sum_{e \in \mathcal{E}} a_{ef} y_e \geq 1 \quad (f \in \mathcal{F}) \quad (8)$$

$$\sum_{e \in \mathcal{E}} b_{ep} y_e = 1 \quad (p \in \mathcal{P}) \quad (9)$$

$$y_e \geq 0 \quad (e \in \mathcal{E}) \quad (10)$$

with:

- \mathcal{E} : Set of legal crew schedules
- c_e : Cost of crew schedule $e \in \mathcal{E}$
- y_e : Variable to denote whether crew schedule $e \in \mathcal{E}$ is included in the minimum-cost subset of the optimal solution or not
- a_{ef} : Binary parameter to denote whether crew schedule $e \in \mathcal{E}$ covers flight $f \in \mathcal{F}$ or not
- b_{ep} : Binary parameter to denote whether crew schedule $e \in \mathcal{E}$ belongs to crew/crew member $p \in \mathcal{P}$ or not

This formulation is equivalent to the LP-relaxation of the original model, but it has only $|\mathcal{F}| + |\mathcal{P}|$ rows instead of $|\mathcal{F}| + |\mathcal{P}| \left(\sum_{r \in \mathcal{R}} |\Pi_r| \right) + |\mathcal{P}| |\mathcal{N}|$.

For crew-member level rescheduling, constraints (8) are replaced by

$$\sum_{p \in \mathcal{P}_q^Q} \sum_{e \in \mathcal{E}_p} a_{ef} y_e \geq d_{fq} \quad (f \in \mathcal{F}, q \in \mathcal{Q}) \quad (11)$$

Since the crew schedule set \mathcal{E} usually is quite large, tabulating all crew schedules is not an option. Instead we define the restricted master problem (RMP) with only a subset of all crew schedules \mathcal{E}' . Solving RMP yields the dual multipliers π_f^F for constraints (8) and π_p^P for constraints (9). We can, thus, define the reduced cost coefficient \tilde{c}_e for variable y_e as

$$\tilde{c}_e = \sum_{p \in \mathcal{P}} b_{ep} \left(\sum_{ij \in \mathcal{A}} \left(c_{pij} - \sum_{f \in \mathcal{F}} \pi_f^F a_{ef} \right) x_{pij}^e - \pi_p^P \right) \quad (12)$$

The simplex criterion requires us to find the minimum reduced cost coefficient

$$\tilde{c}_s = \min_{e \in \mathcal{E}} \tilde{c}_e \quad (13)$$

in order to find a variable y_s to enter the basis. If we can find a y_s with $\tilde{c}_s < 0$, entering crew schedule s into the basis would improve the objective function value and we should thus add the column to RMP. However, if $\tilde{c}_s \geq 0$, the current RMP's optimal solution is also optimal for the full master problem.

Defining the flight subset \mathcal{F}_{ij}^A that contains all flights covered by arc $ij \in \mathcal{A}$, finding such a crew schedule becomes equivalent to solving the following subproblem for each crew/crew member $p \in \mathcal{P}$ as follows:

$$\min \sum_{ij \in \mathcal{A}} \left(c_{pij} - \sum_{f \in \mathcal{F}_{ij}^A} \pi_f^F \right) x_{pij} \quad (14)$$

$$\text{s.t.} \quad \sum_{ij \in \mathcal{A}_{r\pi}} k_{rpij} x_{pij} \leq K_{r\pi} \quad (r \in \mathcal{R}; \pi \in \Pi_r) \quad (15)$$

$$\sum_{ij \in \mathcal{A}} x_{pij} - \sum_{ji \in \mathcal{A}} x_{pji} = s_{ip} \quad (i \in \mathcal{N}) \quad (16)$$

$$x_{pij} \geq 0 \quad (ij \in \mathcal{A}) \quad (17)$$

This problem can be viewed as a resource-constrained shortest-path problem (RCSPP) which can be solved to optimality using dynamic programming (in our case with a modified version of the well-known Bellman-Ford-Moore algorithm for unconstrained shortest path problems). Such an approach also provides the additional benefit that we will only consider candidate columns with $x_{pij} \in \{0, 1\}$ (instead of $x_{pij} \geq 0$, as in (17)). As a consequence, the solution to (7) to (10) provides a tighter LP bound on the optimal integer solution than the LP relaxation of (1) to (5).

6 Solution Method

Since the time needed to obtain a solution is a critical factor in airline irregular operations, we did not attempt to solve the column-generation model directly with a standard MIP-solver such as CPLEX, but instead embedded it in a problem-specific solution method which can deliver optimal solutions within a short time. In the following sections we describe the overall method as well as some key elements. A full description of the solution method can be found in Nissen (2003).

6.1 Algorithmic Scheme

When a disruption occurs at *disturbance point* t^D in an airline's regular crew schedule, we need several parameters in order to find a new schedule. First, we have to determine a *recovery point* t^R from when on the original crew assignments have to be valid again. This decision establishes the *rescheduling period* $\mathcal{T}^R = \{t^D, \dots, t^R\}$. Only flights (though not necessarily all) that take place within this period of time may have their crew assignments changed.

To decide which flights will actually be considered for rescheduling, we require the full set of flights \mathcal{F}^T that the airline's schedule lists within the current scheduling period \mathcal{T} (e.g., the current week or month) for the fleet in consideration and the subset of disturbed flights \mathcal{F}^D . For each flight $f \in \mathcal{F}^T$ we need the following information:

- Original crew assignments \mathcal{P}_f^F
- Departure time t_f^{SF}
- Departure airport ap_f^{SF}
- Arrival time t_f^{TF}
- Arrival airport ap_f^{TF}

This implicitly also defines the set of airports \mathcal{AP} that the airline flies to and from with these flights. We then need to know the set \mathcal{P}^T of all crews/crew members that are available for at least some time within the rescheduling period (i.e., all scheduled crews and the reserve crews) and the subset \mathcal{P}^D of crews/crew members directly affected by the disturbance (i.e., those assigned to the flights from \mathcal{F}^D). For each crew/crew member $p \in \mathcal{P}^T$ we need to know the set of originally assigned flights $\mathcal{F}_p^P = \{f_{p,1}^P, \dots, f_{p,|\mathcal{F}_p^P|}^P\}$, ordered in sequence of their departure times. It is also known how \mathcal{F}_p^P is partitioned into duty periods $dp_{j,p}^P$, with $\mathcal{DP}_p^P = \{dp_{p,1}^P, \dots, dp_{p,|\mathcal{DP}_p^P|}^P\}$.

Furthermore, the set of work rules that have to be observed is required as input data.

The general algorithmic scheme of the presented method for crew rescheduling is summarized in Algorithm 1. It begins by selecting from all available crews a subset of crews \mathcal{P} that can be used to remedy the disturbance. This decision also implies a set of flights

Algorithm 1. Crew Rescheduling: Algorithmic Scheme

```

begin
  input rescheduling period  $T^R$ 
  input set of work rules
  input set of flights  $\mathcal{F}^T$  and subset of disturbed flights  $\mathcal{F}^D$ 
  input set of crews/crew members  $\mathcal{P}^T$  and affected subset  $\mathcal{P}^D$ 
  select crew set  $\mathcal{P}$  and flight sets  $\mathcal{F}^R$  and  $\mathcal{F}^F$ 
   $\mathcal{F} = \mathcal{F}^R \cup \mathcal{F}^F$ 
  generate set of duty periods  $\mathcal{DP}$  and network  $G = (\mathcal{N}, \mathcal{A})$ 
  set upper bound  $UB = \infty$ 
  generate initial set of columns  $\mathcal{E}$ 
  generate columns in branch-and-bound framework
  output new crew schedule
end

```

\mathcal{F}^R that can have their crew assignments changed. Since the solution method is based on duty periods, it is necessary that we always select full duty periods, even though we may allow only some of a duty period's flights to be reassigned. Thus, we will also select a subset of flights \mathcal{F}^F with fixed crew assignments. Most notably this is the case, if a duty period begins before the disturbance point t^D or ends after the recovery point t^R . The crew and flight selection algorithm is described in Section 6.2.

After the set of flights \mathcal{F} and the set of crews/crew members \mathcal{P} for rescheduling have been determined, the next step is to generate the set of all legal duty periods \mathcal{DP} that can be derived from \mathcal{F} . With \mathcal{DP} known, it is possible to generate a time-line network $G = (\mathcal{N}, \mathcal{A})$. The arc set \mathcal{A} is constructed by representing duty periods as arcs and then connecting them with rest period arcs where permitted by the respective regulations. The network is amended with dummy source and sink nodes for the crews, and these are connected to duty period arcs where possible. Creating the network also includes generating the resource sets for those work rules that have to be covered by resources.

The last preparatory step is to generate an initial set of columns as a starting point for the ensuing search for a solution.

After all the necessary data has been created, the actual search for a new crew schedule starts. This is done in a column generation approach with a set partitioning model as the master problem and a resource-constrained shortest path problem for each crew/crew member as subproblems.

6.2 Crew and Flight Selection

To guarantee an optimal solution in terms of rescheduling costs, one would need an infinite rescheduling period T^R and then allow all the flights that the airline's schedule lists within T^R to be rescheduled. However, an infinite recovery period is neither possible nor desirable. But even if the rescheduling period is set to a more realistic period of several hours to a few days, the need for fast solution times usually mandates that not

all the airline's flights within this period be considered, but only a subset \mathcal{F} with a corresponding subset of crews \mathcal{P} . However, this brings with it the possibility that the new schedule derived from \mathcal{F} is suboptimal from a global perspective, or even that no new schedule can be found at all. Hence, a good method for selecting crews and flights for rescheduling that balances fast solution times and low cost schedules is of critical importance.

The method we present here is aimed at selecting a subset of crews according to their capability to cover the flights which are temporarily left uncovered by the disturbance. The method was also designed to be simple and easily adaptable to different situations by changing only a few parameters. Thus, the extent to which crews are selected is governed by the following set of parameters:

- The initial set of candidate flights \mathcal{F}^{cand} (e.g., the subset \mathcal{F}^D of affected flights).
- The iteration parameter j^{CS} determines how many iterations of the selection process are done, with each iteration possibly adding more crews to \mathcal{P} .
- The neighborhood parameter n^{CS} specifies how many flights preceding and succeeding a flight candidate will be checked as further crew/flight candidates.
- The time window parameter t^{CS} indicates the time window around a flight candidate's departure and arrival time that will be examined.

The basic assumption underlying the selection process is that a crew/crew member $p \in \mathcal{P}$ should be considered for inclusion in the problem, if it is available within the rescheduling period at either the departure airport of an affected or uncovered flight $f \in \mathcal{F}^T$ around its departure time or at the arrival airport of flight f around its time of arrival. The crew can then be used to either cover flight f or take over another crew's schedule if that crew is reassigned to cover the flight. Formally this is expressed by defining time windows around the flight's departure and arrival time that take into account the rescheduling period:

$$\begin{aligned}\mathcal{T}_f^{SCS} &= \left\{ \max(t_f^{SF} - t^{CS}, t^D), \dots, \min(t_f^{SF} + t^{CS}, t^R) \right\} \\ \mathcal{T}_f^{TCS} &= \left\{ \max(t_f^{TF} - t^{CS}, t^D), \dots, \min(t_f^{TF} + t^{CS}, t^R) \right\}\end{aligned}$$

and then deriving the crew sets for inclusion in the problem as follows:

$$\begin{aligned}\mathcal{P}_f^{SCS} &= \left\{ p \in \mathcal{P} \mid \begin{array}{l} p \text{ is available at } ap_f^{SF} \text{ for} \\ \text{at least one period during } \mathcal{T}_f^{SCS} \end{array} \right\} \\ \mathcal{P}_f^{TCS} &= \left\{ p \in \mathcal{P} \mid \begin{array}{l} p \text{ is available at } ap_f^{TF} \text{ for} \\ \text{at least one period during } \mathcal{T}_f^{TCS} \end{array} \right\}\end{aligned}$$

Furthermore, we define:

$$\begin{aligned}f_{fp}^{SCS, pred} &= (f' \in \mathcal{F}_p^P \mid t_{f'}^{TF} = \max \{ t_{f''}^{TF} \leq \min \{ t \in \mathcal{T}_f^{SCS} \} \mid f'' \in \mathcal{F}_p^P \}) \\ f_{fp}^{SCS, succ} &= (f' \in \mathcal{F}_p^P \mid t_{f'}^{SF} = \min \{ t_{f''}^{SF} \geq \max \{ t \in \mathcal{T}_f^{SCS} \} \mid f'' \in \mathcal{F}_p^P \}) \\ f_{fp}^{TCS, pred} &= (f' \in \mathcal{F}_p^P \mid t_{f'}^{TF} = \max \{ t_{f''}^{TF} \leq \min \{ t \in \mathcal{T}_f^{TCS} \} \mid f'' \in \mathcal{F}_p^P \}) \\ f_{fp}^{TCS, succ} &= (f' \in \mathcal{F}_p^P \mid t_{f'}^{SF} = \min \{ t_{f''}^{SF} \geq \max \{ t \in \mathcal{T}_f^{TCS} \} \mid f'' \in \mathcal{F}_p^P \})\end{aligned}$$

The selection process (see Algorithm 2) starts with an initial set of candidate flights \mathcal{F}^{cand} (e.g., the subset of affected flights \mathcal{F}^D) and then determines for each flight $f \in \mathcal{F}^{cand}$ the crew/crew member sets \mathcal{P}_f^{SCS} and \mathcal{P}_f^{TCS} . The crews from these two sets are added to the set \mathcal{P}^R . In the same fashion the selection process looks at the n^{CS} flights that precede or succeed flight f in the schedule(s) of the crew(s) assigned to flight f .

The candidate set for the selection process' next iteration is determined by looking at the flights preceding and succeeding \mathcal{T}_f^{SCS} and \mathcal{T}_f^{TCS} , respectively, for each crew. The flights $f_{fp}^{SCS,pred}$ and $f_{fp}^{SCS,succ}$ that directly precede/succeed the time window \mathcal{T}_f^{SCS} in \mathcal{F}_p^P are added to \mathcal{F}^{cand} if they fall at least partially within the rescheduling period. The same is done for the flights $f_{fp}^{TCS,pred}$ and $f_{fp}^{TCS,succ}$ directly preceding/succeeding the time window \mathcal{T}_f^{TCS} in \mathcal{F}_p^P .

This first part of the selection process is repeated j^{CS} times, which allows for a gradual extension of crew set \mathcal{P}^R .

After the crew selection is finished, the two crew/crew member sets \mathcal{P}^D and \mathcal{P}^R are joined to form the set \mathcal{P} . The set of flights \mathcal{F}^R is then derived by allowing all flights within the rescheduling period \mathcal{T}^R that are assigned to a crew/crew member from \mathcal{P} to have their crew assignments changed. Since scheduling is always done in full duty periods, the selection process also produces a set of flights \mathcal{F}^F with fixed crew assignments. These are needed to complete those duty periods from which at least one flight was included in \mathcal{F}^R and which were already begun before the disturbance point t^D or which end after the recovery point t^R . Note that the affected flights in \mathcal{F}^D will be distributed to \mathcal{F}^R and \mathcal{F}^F , depending on when they take place.

Finally, we have to determine the start and end points for each crew/crew member $p \in \mathcal{P}$, i.e., the locations where they first become available and where they have to be located to continue with the part of their old schedule that was not included in the problem, respectively. The former is usually the airport where the first of the crew's selected flights takes off, whereas the latter is typically the airport where the first of the crew's assigned, but not selected flights after the rescheduling period takes off. In both cases the time when they have to be available is at the respective flight's departure time minus the mandatory briefing period. For reserve crews the starting point is their home base at the beginning of the rescheduling period, whereas their end point is under no particular restrictions other than those that the rescheduler may impose, i.e., returning the crew again to its home base.

6.3 Generation of Initial RMP

For the restricted master problem the main issue in implementing a solution method is the way in which the initial RMP is generated. It has to be set up in a way that guarantees a feasible solution so that the dual multipliers π_p^P and π_f^F can be obtained.

To assure satisfaction of constraint (9), one column is added for each crew/crew member with a one in the crew/crew member's corresponding row, a zero in all other rows, and zero cost. If chosen in the optimal solution, it would mean that the respective crew/crew member is not assigned any flights.

Algorithm 2. Crew and Flight Selection

```

begin
  input rescheduling period  $\mathcal{T}^R$ 
  input set of flights  $\mathcal{F}^T$  and candidate subset  $\mathcal{F}^{cand}$ 
  input sets of crews/crew members  $\mathcal{P}^T$  and  $\mathcal{P}^D$ 
  input selection parameters  $j^{CS}, n^{CS}, t^{CS}$ 
   $\mathcal{P}^R = \emptyset$ 
  for  $j = 1$  to  $j^{CS}$  do
     $\mathcal{F}^{check} = \mathcal{F}^{cand}, \mathcal{F}^{cand} = \emptyset$ 
    for each  $f \in \mathcal{F}^{check}$  do
       $\mathcal{P}_f^F = \{p \in \mathcal{P} \mid p \text{ is assigned to } f\}$ 
      if  $\mathcal{P}_f^F = \emptyset$  then
         $\mathcal{P}^R = \mathcal{P}^R \cup (\mathcal{P}_f^{SCS} \cup \mathcal{P}_f^{TCS})$ 
      else
        for each  $p \in \mathcal{P}_f^A$  do
           $n = \text{sequence position of } f \text{ in } \mathcal{F}_p^P$ 
          for each  $f' \in \{f_{p,n-n^{CS}}^P, \dots, f_{p,n+n^{CS}}^P\}$  do
             $\mathcal{P}^R = \mathcal{P}^R \cup (\mathcal{P}_{f'}^{SCS} \cup \mathcal{P}_{f'}^{TCS})$ 
          end for
        end for
      end if
    end if
    for each  $p \in (\mathcal{P}_{f'}^{SCS} \cup \mathcal{P}_{f'}^{TCS})$  do
      add  $f_{fp}^{SCS,pred}, f_{fp}^{SCS,succ}, f_{fp}^{TCS,pred}, f_{fp}^{TCS,succ}$  to  $\mathcal{F}^{cand}$ , if they fall
      at least partially within  $\mathcal{T}^R$  and have not been in  $\mathcal{F}^{cand}$  before
    end for
  end for
end for
 $\mathcal{P} = \mathcal{P}^D \cup \mathcal{P}^R$ 
 $\mathcal{F}^R = \{f \in \mathcal{F}^T \mid \{t_f^{SF}, \dots, t_f^{TF}\} \in \mathcal{T}^R \wedge (\exists p \in \mathcal{P} \mid f \in \mathcal{F}_p^P)\}$ 
 $\mathcal{F}^F = \{f \in \mathcal{F}^T \mid \{t_f^{SF}, \dots, t_f^{TF}\} \notin \mathcal{T}^R \wedge$ 
    $(\exists f' \in \mathcal{F}^R, p \in \mathcal{P}, dp \in \mathcal{DP}_p^P \mid f, f' \in dp)\}$ 
  for each  $p \in \mathcal{P}$  do
    determine time  $t_p^{SP}$  and location  $ap_p^{SP}$  of earliest availability
    determine time  $t_p^{TP}$  and location  $ap_p^{TP}$  of latest recovery
  end for
  output set of reschedulable flights  $\mathcal{F}^R$ 
  output set of flights  $\mathcal{F}^F$  with fixed crew assignments
  output set of crews/crew members  $\mathcal{P}$ 
  output time  $t_p^{SP}$  and location  $ap_p^{SP}$  of earliest availability for each  $p \in \mathcal{P}$ 
  output time  $t_p^{TP}$  and location  $ap_p^{TP}$  of latest recovery for each  $p \in \mathcal{P}$ 
end

```

To maintain the necessary flight coverage of constraint (8), one column is added for each flight with a one in the flight's corresponding row, a zero in all other rows, and high cost. Such a column assigns the flight to no crew at all, so that including it in the optimal solution would mean that the flight has to be cancelled. In the case of crew-member-level-based rescheduling with multiple qualification groups, we will have a column with the values δ_{fq} in the flight's corresponding rows, and zeros in all other rows. This assures that a flight is either assigned a full crew complement or none (i.e., it is cancelled), but never a partial crew complement. This approach is comparable to applying the M-method to obtain a feasible solution.

Having this set \mathcal{E}^{Min} of $|\mathcal{P}| + |\mathcal{F}|$ columns in the RMP guarantees feasibility in that it is always possible to not carry out a flight and having a crew/crew member not carry out any duties at all. Note, that in a branch-and-bound framework these columns have to be kept in the RMP throughout the search tree to retain feasibility.

However, having only this minimum set of initial columns yields an arbitrarily bad initial solution by cancelling all flights and having all crews/crew members spend idle time. The original schedule prior to the disturbance allows us to provide a better starting point by adding another column for each crew/crew member from the set of unaffected but reschedulable crews/crew members \mathcal{P}^R , representing its original schedule for the rescheduling period \mathcal{T} . It has a one in each of the rows of constraint (8) for the originally assigned flights, and also a one in the row of restriction (9) for the respective crew/crew member. Since in this column no flights were reassigned to the crew/crew member, the column's cost is set to zero.

We can further improve the flight coverage in the initial solution (and thus provide a better upper bound) by trying to include a column for each affected crew/crew member $p \in \mathcal{P}^D$. This can be done by attempting to build a new schedule from the flights originally assigned to p by finding the earliest possible match-up point after the crew's schedule disturbance (if it exists).

For larger problems with many crews without a start column (i.e., reserve crews) and thus many initially uncovered flights, convergence may become a problem. To counter this, we provided the possibility to add start columns for such crews by solving the SPP once to gain reduced cost information, then solving the RCSPPs only for those crews while forbidding that flights be covered by more than one crew in this procedure. This '*selective first column generation round*' leads to higher initial flight coverage, thus providing a better upper bound and speeding up convergence. However, only larger problems benefit from this procedure.

6.4 Branch-And-Bound Framework

If the column generation algorithm can find no more columns that price out to enter the basis and the LP solution does not satisfy the integrality property, we need to employ a branch-and-bound algorithm to obtain an integer solution.

Applying a standard branch-and-bound procedure to the master problem over existing columns is unlikely to find an optimal, or even feasible, solution to the original problem. In addition, conventional branching on the variables y_e is ineffective. It would mean

either fixing the flight assignments of column e or forbidding that column. While fixing the assignments is easily implemented, forbidding a certain schedule is difficult, because it represents a particular solution to one of the subproblems. It is quite likely that the next time the subproblem is solved, the optimal solution is precisely the one represented by y_e . In that case it would be necessary to find the second best solution to the subproblem. At depth l in the branch-and-bound tree we may need to find the l -th best solution, in effect destroying the structure of the pricing problem.

Thus, we have to devise a branching rule that is compatible with the pricing problem. This means that we must be able to modify the subproblems so that columns that are infeasible due to the branching decisions will not be generated and the column generation subproblem will remain tractable.

Furthermore, the branch-and-bound framework must allow us to generate additional columns not only at the root node, but throughout the search tree, since it is very unlikely that all the columns covered in the optimal integer solution will have already been generated at the root node.

We will use a branching rule called *branch-on-follow-on* which was proposed by Vance et al. (1997) for crew scheduling and is an extension of a rule proposed by Ryan, Foster (1981) for set partitioning problems. It originates from the observation that in a fractional solution two columns e and e' must exist with y_e and $y_{e'}$ both fractional such that they exhibit the Ryan-Foster submatrix shown in Figure 3 for at least two rows (flights) r and s . When using *branch-on-follow-on*, we are looking for two such flights that appear consecutively in the corresponding schedule of at least one fractional column. On the first branch we will then force them to appear consecutively, whereas on the second branch we will forbid them to appear consecutively in any schedule/column. Vance et al. (1997) show that such a pair always exists if the LP solution is fractional.

Figure 3: Ryan-Foster Submatrix

	e	e'
r	1	1
s	0	1

We use a binary branching tree where the root node is the initial problem given by formulation (7) to (10). New nodes are created by adding branching decisions to both the master problem and the subproblem. Branching is done as a depth-first search.

7 Computational Experience

In order to evaluate the efficiency of the model and the solution approach, a series of computational experiments was conducted. The experimental design and key results are presented in this section. A detailed description of all results can be found in Nissen (2003).

The computational experiments were conducted on a PC with an Athlon 1200 MHz CPU and 512 MB RAM, running Windows 2000 Professional. The initial crew and flight selection was done in Access 2000 with Visual Basic. The results were output into a text file which was then read by the remainder of the solution method. It was programmed in ANSI C and compiled with the Visual C++ 6.0 compiler. The LP relaxations of the master problem were solved using CPLEX, version 7.0.0 (CPLEX Optimization Inc. 2000). All time measurements are given in CPU seconds.

7.1 Experimental Design

To test the solution method repeatedly in a wide variety of typical rescheduling scenarios, we did not use actual rescheduling cases from an airline, but instead generated them with a test instance generator.

The data basis was the dated flight schedule of a major European carrier. Due to confidentiality restrictions for personnel data in the airline's union agreements, the airline's actual crew schedule was not available, so we used the flight schedule to generate two crew schedules: one schedule for short-haul flights and one for medium-haul flights. Both have a hub-and-spoke structure with only one hub and cover a period of seven days – from Monday to Sunday. The short-haul schedule has eight routes and covers 450 flights from one fleet, whereas the medium-haul schedule has 35 routes and covers 927 flights from one fleet. The schedules were generated to comply with the German regulatory framework. The period length was set to 5 minutes.

We did not consider long-haul schedule problems in our experiments, as they are characterized by flights that typically do not operate daily, so that flight connections are sparse. Thus, for these problems the main challenge is to find a good method for selecting potential deadhead flights (e.g., the deadhead selector described in Barnhart et al. (1995)), whereas a fast solution method is usually only of secondary concern.

We defined the following list of typical rescheduling scenarios for which we wanted to test the solution method:

- Scenarios 1 – 3: Incidental delay of a single flight for 30, 60 and 120 minutes, respectively
- Scenario 4: Cancellation of a single flight
- Scenario 5: Equipment swap: Two flights from different fleets that take place on the same route at roughly the same time are swapped between the fleets. For each single-fleet schedule this means that one flight is cancelled and a new flight is inserted at approximately the same time.
- Scenario 6: Addition of a new flight to the schedule
- Scenarios 7 – 9: Delay of five flights, each for 30, 60 and 120 minutes, respectively. The flights are scheduled to take off from different airports, but all within a time window of 60 minutes.

- Scenarios 10 – 12: Delay of five flights, each for 30, 60 and 120 minutes, respectively. All flights are scheduled to take off from the same airport within a time window of 120 minutes.
- Scenarios 13 – 14: Hub airport is closed for 60 and 120 minutes, respectively, so that all flights arriving at or leaving from the hub during that period have to be either delayed, cancelled or rerouted.

For each scenario, we generated five test cases for each of the two schedules and then applied the solution method. All schedule disturbances were imposed on Tuesday, i.e., the second day of the original schedule. This day was chosen so that on the one hand the crews have had some prior workload before the disturbances occurred, and on the other hand there would be enough scheduled days left to compare the effects of different rescheduling periods.

The test cases were tested with various parameter settings regarding the rescheduling period, the selective first column generation round, the crew and flight selection method.

A full test design in which all parameter combinations would have been analyzed was not possible because it would have resulted in several thousand test instances for each case. So we chose a step-wise approach, in which we first investigated the recovery period to make a good choice, and then examined the other parameter groups separately on this rescheduling horizon for their effects on overall solution times and/or solution quality. Finally, we derived a ‘best’ parameter set from these results, and conducted further tests to obtain more insights into the algorithm’s performance.

The rescheduling objective was to first cover as many flights as possible, and then to cover these flights with as few reassignments as possible. Thus, the cost for the flight coverage column in the initial RMP was set high enough that flight coverage would be the foremost objective regardless of the number of crew reassignments necessary. In the columns generated in the subproblems, each flight assigned to another crew as its original crew incurred costs of 1.

7.2 Recovery Horizon

The first parameter to be investigated was the recovery horizon. Different horizon lengths were compared for their effect on the solution quality and solution time with the results obtained from the longest recovery period serving as the reference.

Table 5 shows the results for both schedules. Since computing relative gaps between solutions would depend on the relative weights given to uncovered flights and schedule changes, we used instead the percentages of cases where the solution value exactly equaled the reference value from the full length recovery period (‘Absolute Quality’ row) and the percentage where the solution at least left no more flights uncovered (‘Change Quality’ row, in brackets). The reason for this second comparison is that a schedule that is not exactly as good as the ‘best’ schedule may still be acceptable if it at least manages to cover the same number of flights.

For the short-haul schedule a recovery period of 48 hours was deemed to be the best compromise between solution quality and time, whereas for the medium-haul schedule a

Table 5: Solution Quality and Times for Different Recovery Horizons

	12h	24h	48h	60h	96h	Full	
Short Haul	25.7%	44.3%	85.7%	—	100 %	100%	Abs.Qual.
	(34.3%)	(58.6%)	(98.6%)		(100%)	(100%)	(Chg.Qual.)
	0.3	1.0	5.3		38.7	84.1	Avg. Time
	(0.1)	(0.7)	(0.8)		(11.7)	(34.1)	(Std.Dev.)
Medium Haul	—	42.9%	72.9%	100%	100%	100%	Abs.Qual.
		(52.9%)	(88.6%)	(100%)	(100%)	(100%)	(Chg.Qual.)
		6.8	29.7	57.3	163.2	344.9	Avg. Time
		(4.0)	(7.9)	(16.1)	(91.2)	(212.9)	(Std.Dev.)

60 hour recovery horizon produced the best results. For both schedules we also continued to use the full horizon as a reference in all further tests.

7.3 Selective First Column Generation Round

Table 6 shows the results of the selective first column generation round compared to not using the additional algorithm.

As was already said in Section 6.3, only larger problem would benefit from the procedure. So it is no surprise that the short-haul schedule solution times on the 48-hour recovery horizon even increased and that for the full horizon the benefits were too small to justify use of the procedure.

However, for the medium-haul schedule using the selective first column generation round improved solution times significantly: For the 60-hour problems it reduced computation time by about 15%. Whereas scenarios 1–6 where only a single flight was affected benefited only marginally or even had small increases in computation time, scenarios 7–14 with multiple flights affected showed significant benefits. The most drastic improvement was achieved for scenario 14 where computation time was reduced by 45% on average.

For the full horizon problems degeneracy and slow convergence played a significant role, and thus the algorithm’s savings reached on average 88%. Again, the scenarios where only one flight was affected showed mostly moderate benefits, but for the other scenarios the savings were immense. For scenarios 13 and 14 solving the problems without the selective first column generation round would have taken on average several hours, thus exceeding the duration of the whole disturbance. Using the algorithm reduced solution times by 97% and 94%, respectively, thus bringing them down to an acceptable level.

7.4 Crew and Flight Selection

Since the best opportunity to trim down solution times lies in reducing problem size, Section 6.2 introduced a crew and flight selection method that allows to find a solution with only a subset of all available crews and flights.

We tested the method with two different types of initial candidate flight sets \mathcal{F}^{cand} :

Table 6: Solution Times with/without Selective First Column Generation Round

	60h		Full		
	with	without	with	without	
Short Haul	5.4 (0.9)	5.3 (0.8)	78.9 (69.8)	84.1 (34.1)	Avg. (Std.Dev.)
Medium Haul	57.3 (16.1)	67.4 (31.0)	344.9 (212.9)	2,769.0 (7,258.3)	Avg. (Std.Dev.)

1. \mathcal{F}^{cand} includes all flights from \mathcal{F}^D
2. \mathcal{F}^{cand} includes all flights within the rescheduling period that cannot be covered by a start column generated with the ‘match-up point’ method from Section 6.3

The evaluation combined the two candidate flight set types with all combinations of the following values for the three parameters j^{CS}, n^{CS}, t^{CS} :

- $j^{CS} \in \{1, 2, 3\}$
- $n^{CS} \in \{0, 1, 2\}$
- $t^{CS} \in \{60 \text{ min}, 120 \text{ min}\}$

We investigated the effects on problem size, solution quality (in percent of ‘best’ solutions found) and solution time on problems with a 48-hour/60-hour recovery period and on those with the full horizon. Since the results for the ‘uncovered flights’ candidate set were more robust with respect to parameter changes, it was deemed to be better suited for rescheduling, and, thus, only those results are reported here.

Table 7 shows the results for the two schedules. For each schedule/horizon we report the results for the best parameter combination that yielded the same solution quality as no selection would. As a reference we also report the results obtained without the selection method. We indicate the problem sizes by giving the average and standard deviation (in brackets) for the number of flights, nodes and arcs. The last column states the average and standard deviation (in brackets) for the solution time.

For the short-haul schedule the savings are moderate: For the 48-hour horizon, the average number of included flights is reduced by almost 7%, which in turn reduces the arc set by about 15% and solution time by approximately one quarter. For the full horizon, the selection achieves reductions in the average number of included flights by 6%, leading to a reduction in the arc set of 12% and solution times by approximately 13%.

For the medium-haul schedule bigger savings were achieved: For the 60-hour horizon, the average number of included flights is reduced by about 15%, which in turn reduces the arc set by about 25% and solution time by almost 40%. For the full horizon, reduction in the flight set averages 15%, leading to reductions in the arc set of about 23%, and solution times by approximately 30%.

Table 7: Crew and Flight Selection by Uncovered Flights

	Recovery Period	Selection (j^{CS}, n^{CS}, t^{CS})	Flights	Nodes	Arcs	Total	
Short Haul	48 h	(1,1,120)	282.5 (28.1)	473.5 (59.7)	11,221.4 (2,349.4)	3.9 (1.7)	Avg. (Std. Dev.)
		No Selection	301.5 (7.8)	514.0 (11.1)	13,087.2 (602.3)	5.3 (0.8)	
	Full	(1,1,120)	391.8 (37.1)	948.8 (106.6)	37,239.4 (7,296.7)	73.1 (42.5)	
		No Selection	415.7 (15.6)	1,016.7 (47.3)	42,293.0 (3,869.3)	84.1 (34.1)	
Medium Haul	60 h	(1,1,60)	543.3 (100.5)	971.5 (188.4)	32,629.5 (10,412.2)	34.9 (21.8)	
		No Selection	645.4 (30.6)	1,156.9 (81.7)	43,646.8 (6,076.3)	57.3 (16.1)	
	Full	(1,1,60)	730.7 (142.1)	1,686.6 (336.2)	79,650.9 (25,484.0)	239.0 (239.8)	
		No Selection	859.6 (33.9)	1,982.2 (92.2)	103,384.2 (10,246.0)	344.9 (212.9)	

7.5 Branch-And-Bound Framework

Another important aspect of the algorithm is the quality of the branch-and-bound framework. Fortunately, however, the solution method rarely produces fractional solutions at the root node. With the various recovery horizons and crew and flight selection schemes, we ran a total of 5,390 experiments on the short-haul schedule, and only 18 of those produced a fractional solution at the root node. Two of those belong to the 24-hour test instances that produced solutions which were infeasible in practice.

For the remaining 16 cases where branching occurred, no case had an optimality gap between the LP lower bound and the optimal integer solution. Since there was no optimality gap, the node where the optimum was found was also always the last node to be visited. By comparing the depth of the node where the optimal solution found and the maximum depth reached during the branching process, we can see that the optimum was typically found rather deep in the search tree. In fact, only in 2 of the 16 instances, backtracking did occur.

For the medium-haul schedule, we saw even less branching: Only one of the 5,740 experiments required branching, and that was an instance that also showed slow convergence. Just as for the short-haul branching instances, this single case had no optimality gap between the LP lower bound and the optimal integer solution. It required 8 branching nodes, with the maximum depth also being 8 nodes.

The fact that the larger instances – when compared to the short-haul instances – required branching even less frequently is most likely due to the use of the selective first column generation round. For neither schedule branching was necessary with the respective ‘best’

Table 8: Algorithm Performance Details

	Init Time	RCSP		LP				Total Time	
		No.	Time	Columns	Rows	No.	Time		
Short	2.8	181.6	1.1	385.7	323.0	4.3	0.008	3.9	Avg. (Std. Dev.)
Haul	(1.1)	(89.5)	(0.7)	(52.7)	(33.7)	(1.8)	(0.010)	(1.7)	
Medium	29.9	226.8	4.9	670.4	607.4	6.9	0.020	34.9	
Haul	(18.3)	(160.5)	(4.7)	(129.9)	(112.9)	(4.0)	(0.022)	(21.8)	

parameter constellation which is described in Section 7.6

7.6 Algorithm Details for Best Parameter Set

The experiments described in the previous section allow us to formulate a ‘best’ parameter constellation for each of the two schedules.

For the short-haul schedule problems this means using a 48-hour recovery horizon, crew and flight selection with the ‘uncovered flight’ candidate set and $j^{CS} = 1, n^{CS} = 1, t^{CS} = 120$ min. The medium-haul schedule problems were best solved with a 60-hour recovery horizon, crew and flight selection with the ‘uncovered flight’ candidate set and $j^{CS} = 1, n^{CS} = 1, t^{CS} = 60$ min, and use of the selective first column generation round.

We will close our analysis with a look at how the total computation time is composed, i.e., how the computation time is spread over the various elements of the solution method for these two parameter constellations.

Table 8 shows averages and standard deviations for several key indicators about the algorithm’s computational performance for the two schedules with the respective best parameter set: The second column gives the time that was needed to initialize the problem, i.e., generate duty periods and network. The next two columns show the performance of the multi-label algorithm for the RCSP by indicating the number of problems solved and the total time spent on it. The following three columns describe the solution of the master problems, i.e., the part that was solved by CPLEX. They state the number of columns in the LPs, the number of LPs solved and the time expended. Finally, the last column gives the total computation time.

As we can see, a significant portion of the overall computation time is spent on generating the actual problem first. However, as the time needed to generate the problem will to a large extent depend on the way the crew, flight and schedule data is initially provided (i.e., by the airline’s scheduling/rescheduling system), we refrained from optimizing this part of the algorithm. If the solution method were to be integrated into an airline’s scheduling/rescheduling system, the data generation could thus be further improved. In fact, some of the data can already be generated before a rescheduling situation actually occurs.

When looking at the column generation part, it is quite obvious that the LPs are easily solved by CPLEX. Most of the column generation time is spent for the RCSPs. Although

for the short-haul schedule each shortest path problem took on average only 5.8 ms to solve, the multitude of problems that have to be solved adds up. For each LP we have to solve the RCSPP $|\mathcal{P}|$ times, which for the best parameter set meant solving it on average 40.5 times per LP. This shows how essential it is to keep the number of crews selected for inclusion in the problem as small as possible. For the medium-haul problems, the time spent on solving the RCSPPs increased by a factor of about 4.5 compared to the short-haul problems, whereas the LP solution time increased only by a factor of about 2.5. The increase in the time needed for the shortest path problem can be attributed to the increase in network size. The average solution time for a subproblem increased to 19.3 ms. Since the average number of crews per problem also increased to 64.1, the increase in the RCSPP time should have been even higher, but the selective column generation round resulted in the fact that on average the RCSPP was solved only 31.7 times per LP.

8 Summary

In this work we presented a new duty-period-based formulation for the airline crew re-scheduling problem that uses a new type of resource constraints to efficiently cover the various labor regulations and that is tailored to the needs of European airlines, as well as a solution method based on branch-and-price.

The predominance of fixed crew salaries in Europe offered the opportunity to part with the reliance of previous approaches on repairing the several-day-long pairings and to work with one-day duty periods instead. This has the advantage that it makes shorter rescheduling horizons and thus smaller problems possible. Thus, we stated a model for crew rescheduling based on duty periods. Since it does not use pairings, it allows us to integrate the two steps of crew pairing and assignment from the initial scheduling process into one.

The solution method was then exhaustively tested on various rescheduling scenarios, each with several distinct cases. One of the objectives of this study was to show the effect that different parameter settings for, e.g., rescheduling horizon or crew and flight selection, had on solution quality and time. Especially solution time is a critical factor and the results showed that the solution method is capable of providing solutions within the short period of time available to a rescheduler after a disturbance occurs.

References

- Anbil, R., Gelman, E., Patty, B., Tanga, R. (1991): Recent advances in crew-pairing optimization at American Airlines. *Interfaces*, Vol. 21, No. 1, pp. 62–74.
- Bard, J.F., Yu, G., Argüello, M.F. (2001): Optimizing aircraft routings in response to grounding and delays. *IIE Transactions*, Vol. 33, pp. 931–947.
- Barnhart, C., Hatay, L., Johnson, E.L. (1995): Deadhead selection for the long-haul crew pairing problem. *Operations Research*, Vol. 43, No. 3, pp. 491–499.

- Bixby, R.E., Gregory, J.W., Lustig, I.J., Marsten, R.E., Shanno, D.F. (1992): Very large scale linear programming: A case study in combining interior point and simplex methods. *Operations Research*, Vol. 40, No. 5, pp. 885–897.
- Böttcher, J., Drexl, A., Kolisch, R., Salewski, F. (1999): Project scheduling under partially renewable resource constraints. *Management Science*, Vol. 45, No. 4, pp. 543–559.
- Campbell, K.W., Durfee, R.B., Hines, G.S. (1997): FedEx generates bid lines using simulated annealing. *Interfaces*, Vol. 27, No. 2, pp. 1–16.
- CPLEX Optimization Inc. (2000): Using the CPLEX callable library. CPLEX Optimization, Incline Village, Nevada.
- Day, P.R., Ryan, D.M. (1997): Flight attendant rostering for short-haul airline operations. *Operations Research*, Vol. 45, No. 5, pp. 649–661.
- Desaulniers, G., Desrosiers, J., Dumas, Y., Marc, S. Rioux, B., Solomon, M.M., Soumis, F. (1997): Crew pairing at Air France. *European Journal of Operational Research*, Vol. 97, pp. 220–244.
- Gamache, M., Soumis, F., Villeneuve, D., Desrosiers, J., Gélinas, É. (1998): The preferential bidding system at Air Canada. *Transportation Science*, Vol. 32, No. 3, pp. 246–255.
- Gamache, M., Soumis, F., Marquis, G. (1999): A column generation approach for large-scale aircrew rostering problems. *Operations Research*, Vol. 47, No. 2, pp. 247–262.
- Graves, G.W., McBride, R.D., Gershkoff, I., Anderson, D., Mahidhara, D. (1993): Flight crew scheduling. *Management Science*, Vol. 39, No. 6, pp. 736–745.
- Hoffman, K.L., Padberg, M. (1993): Solving airline crew scheduling problems by branch-and-cut. *Management Science*, Vol. 39, No. 6, pp. 657–682.
- Irrgang, M.E. (1995): Airline irregular operations. In: Jenkins, D., Ray, C.P. (eds.), *Handbook of airline economics (1st edition)*, McGrawHill Aviation Week Group, New York, pp. 349–365.
- Jarrah, A.I.Z, Diamond, J.T. (1997): The problem of generating crew bidlines. *Interfaces*, Vol. 27, No. 4, pp. 49–64.
- Lavoie, S., Minoux, M., Odier, E. (1988): A new approach for crew pairing problems by column generation with an application to air transportation. *European Journal of Operational Research*, Vol. 35, pp. 45–58.
- Lettovský, L., Johnson, E.L., Nemhauser, G.L. (2000): Airline crew recovery. *Transportation Science*, Vol. 34, No. 4, pp. 337–348.
- Nissen, R. (2003): Airline crew rescheduling. *PhD Dissertation*, Christian-Albrechts-University Kiel.
- Rosenberger, J.M., Johnson, E.L., Nemhauser, G.L. (2003): Rerouting aircraft for airline recovery. *Transportation Science*, Vol. 37, No. 4, pp. 408–421.

- Ryan, D.M., Foster, B.A. (1981): An integer programming approach to scheduling. In: Wren, A. (ed.), *Computer scheduling of public transport. Urban passenger vehicle and crew scheduling*, North-Holland, Amsterdam, pp. 269–280.
- Stojković, M., Soumis, M. (1998): The operational airline crew scheduling problem. *Transportation Science*, Vol. 32, No. 3, pp. 232–245.
- Vance, P.H., Atamtürk, A., Barnhart, C., Gelman, E., Johnson, E.L., Krishna, A., Mahidhara, D., Nemhauser, G.L., Rebello, R. (1997): A heuristic branch-and-price approach for the airline crew pairing problem. *Working Paper*, Auburn University.
- Wedelin, D. (1995): An algorithm for large scale 0-1 programming with application to airline crew scheduling. *Annals of Operations Research*, Vol. 57, pp. 283–301.
- Wei, G., Yu, G., Song, M. (1997): Optimization model and algorithm for crew management during airline irregular operations. *Journal of Combinatorial Optimization*, Vol. 1, No. 3, pp. 305–321.
- Yan, S., Lin, C.-G. (1997): Airline scheduling for the temporary closure of airports. *Transportation Science*, Vol. 31, No. 1, pp. 72–82.
- Yu, G., Argüello, M., Song, G., McCowan, S.M., White, A. (2003): A new era for crew recovery at Continental Airlines. *Interfaces*, Vol. 33, No. 1, pp. 5–22.