Decision Support Systems for Airline Crew Recovery

Yufeng Guo
Faculty of Business Administration,
Economics and Business Computing
University of Paderborn

A thesis submitted for the degree of
doctor rerum politicarum
(*Dr. rer. pol.*)

Paderborn, April 2005

to my loving parents

Acknowledgements

First of all, I would like to express my gratitude to my supervisor Professor Dr. Leena Suhl for her guidance and unwavering support that paved the path to the success of the present thesis. I am deeply grateful to her for introducing me to this field and providing great assistance to complete this doctoral thesis. And I would like to extend my grateful thanks to my co-supervisor Professor Dr. Wilhelm Dangelmaier and committee member Professor Dr. Hans Kleine Büning for their helpful advice and reviewing the thesis.

My sincere thanks also go to Professor Dazhe Zhao at Northeastern University in China for introducing me to this international PhD program and her guidance that is far beyond the scope of scientific research.

I would like to thank the *International Graduate School Dynamic Intelligent Systems* at the University of Paderborn for offering me the fellowship and providing financial support to attending several academic conferences from which my research greatly benefited. And I would also like to express my appreciation to the graduate school team, especially Dr. Eckhard Steffen and Astrid Canisius.

I am indebted to my (ex-)colleagues at the DS&OR Lab at the University of Paderborn for their kind support in many ways. Especially, I am grateful to Professor Dr. Taïeb Mellouli for fruitful discussions. Moreover, I would like to thank my colleague and friend Markus P. Thiel who shared an office with me. I enjoy the great cooperation with him and fully appreciate his warm help throughout the past three years.

Furthermore, I wish to express my sincere thanks to Mr. James Harrop who carefully proofread the thesis. He found all the mistakes I made, which was certainly a great help to a non-native writer like me.

I would like to acknowledge the extraordinary support of my family in China, particularly my parents and my brother without whom this work would never be possible. In Chinese: 感谢父亲、母亲和哥哥在过去三年里对我的支持与鼓励！

Finally, my special thanks go to my girlfriend Qiao Chen for her constant support and patience during the preparation of this thesis. I am very thankful for the encouragement and comfort she gave me, which allowed me to concentrate on this work.

Yufeng Guo

Paderborn, April 2005

Abstract

Within the airline industry's complex operational environment, any disturbance to normal operations has dramatic impact, and usually imposes high additional costs. Because of irregular events during day-to-day operations, airline crew schedules can rarely be operated as planned. When disruptions occur, crew schedules are affected due to the resulting infeasible flight schedule and improper assignments. Therefore, airlines need to recover disrupted crew schedules as soon as possible, and minimize the extra cost as well as the impact on subsequent operations.

The task of the airline crew recovery is to obtain one or more reasonable, perfectly optimal, recovery solutions from current disruptions, which has to be achieved within an acceptable period of time. The final solutions are optimized in terms of the amount of additional operational costs and variations from the original planned schedule.

In this thesis, we develop a decision support system that incorporates exact optimization methods and several dedicated heuristics to solve real-life airline crew recovery problems in the setting of European airlines. To solve such a problem, a column generation method and a genetic algorithm based heuristic are proposed and tested. The proposed solution methods are customized with a dedicated setting of parameters, which forms a set of strategies to deal with different disrupted situations. Furthermore, a so called strategy mapping procedure is developed to assist airline coordinators in recovering crew schedules more effectively by investigating the given disruption and proposing a suitable strategy with a proper solution method.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Passenger airlines operate their business in an extremely complex environment. In their daily operations many essential elements are deeply involved, ranging from sophisticated machines to highly skilled humans. Taking information technology (IT) systems as examples, airlines deploy a wide spectrum of tools and software to assist their normal operations, such as reservation systems, revenue management systems, tracking systems, scheduling systems, operations control systems, etc. Each such system is highly sophisticated and has a great impact upon the overall performance of an enterprise.

The increasingly competitive domestic and global markets make it even more difficult to maintain a lucrative and growingly profitable business. To improve their core competence, airlines invest a large amount of time and money in carrying out research which, in turn, supports their business in various ways. During the past decades the airline industry has attracted great attention of many researchers from various areas. Consequently, current airline industry is able to provide the passenger transportation service experienced today by virtue of successful applications of many emerging techniques. However, there still is much room left for airlines to improve their performance.

Generally, the performance of an airline is subject to various factors, both internal and external. For example, the oil price, as an external influencing factor, has economically a dramatic impact as one may observe nowadays. In contrast, internal factors comprise many issues, such as company culture, marketing strategy, human resource management etc., which contribute to the overall efficiency as well.

The essential product provided by a scheduled passenger airline is the *flight schedule* which consists of a set of flights between two or more airports, where each flight is conducted under a given flight number. In order to provide such a public transportation service, many planning processes have to be carried out to manage various types of resources distinctly, such as flight scheduling, crew scheduling, gate scheduling, and so on.

However, in practice airline schedules often cannot be carried out as planned due to many types of unavoidable disturbances. Therefore, airlines need to efficiently manage frequent disruptions and recover their disturbed schedules during irregular operations. Concerning crew schedules, airlines have to find an updated crew schedule with respect to given disruptions and other goals such as cost saving, disturbance minimization, and so on. Because of the complexity of such a task, it is imperative to investigate the recovery process and to establish a system which incorporates a bundle of dedicated problem solving techniques.

This thesis is motivated through the fact that airlines perform their disruption management mostly manually, although computer-based decision support techniques exist that could be used in order to improve the decision quality. However, disruption management tasks are very complex and partially not well structured, so that it is hardly possible to provide optimal solutions in practical disrupted situation. As a matter of fact, it is not clear what is an optimal solution because usually many considerations have to be taken into account simultaneously, and the solution to be adopted in practice is often a compromise taking different goals into account. Furthermore, only during recent years mathematical optimization techniques, software, and hardware have been developed so far that it is possible to try to solve practical disruption problems in airlines.

Because there are many types and sizes of disruptions and decision support techniques to approach them as well as many different goals, it is unlikely that there will be one single method able to solve all airline disruption problems. Thus, a decision support system for airline crew recovery should integrate several algorithmic techniques and manual solution processes of human dispatchers. Furthermore, it should provide a first classification to coordinators in order to determine a suitable solution technique to try first. The system should be usable through an intuitive graphical user interface, and it should be fast in providing a solution, because there is no time to wait in a disrupted situation.

Thus, the goal of the thesis is to improve the current situation at least in small and medium-size airlines through a basic decision support system that integrates several solution methods and chooses them according to classification criteria to be developed within the thesis. After this goal has been achieved, one may proceed towards larger airlines in subsequent research projects.

Prior to the discussion of the problem examined in this work (Section 1.2), a short overview is given about airline planning and the crew management issue in Section 1.1. In Section 1.3 various decision support systems (DSS) that are commonly applied to schedule crew in airlines are introduced briefly. Finally the structure of the thesis is given in Section 1.4.

1.1 Airline Planning and Crew Management

Schedule generation is one of the most elaborate tasks that an airline carries out throughout its operations, as it includes many complex sub-steps. Basically, the overall scheduling process can be composed as a sequence of the following steps (based on the airline scheduling process proposed by Suhl, 1995):

- Block and ground time estimation
- Demand estimation
- Network planning
- Capacity planning
- Fleet assignment
- Aircraft routing
- Flight scheduling
- Crew scheduling
- Tail assignment

- Ground operation scheduling
- Operational rescheduling

Crew management plays a crucial role within the scheduling process, as the cost for managing crew constitutes the second largest expense of an airline after fuel consumption. Unlike other types of expenses, crew costs fall into one of the internal groups of factors that affect an airline's actual revenue. Most importantly, crew costs are relatively controllable by the airline itself. It hence implies great potential to boost revenues by establishing efficient crew management systems.

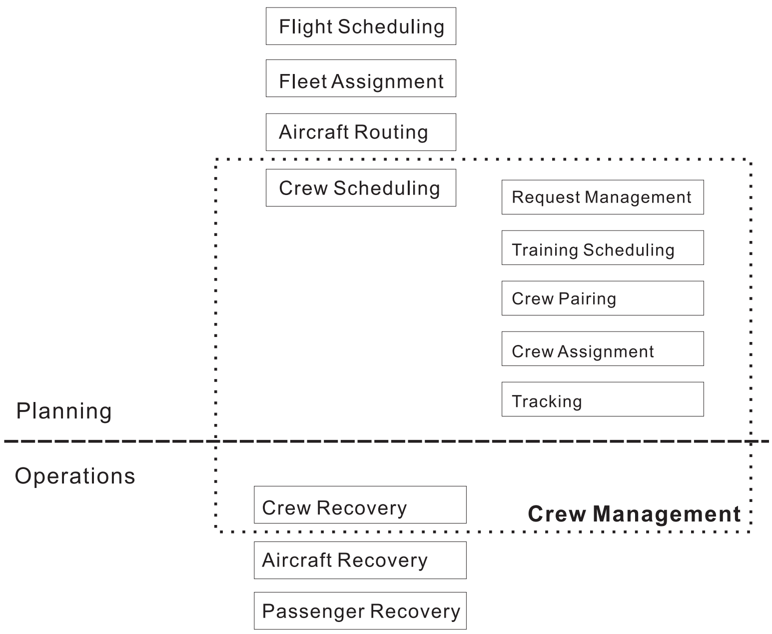


Figure 1.1: Crew management in airline schedule planning process

As one can see in Fig. 1.1, the crew management issue covers two main phases: planning and operations. In the planning phase, the *airline crew scheduling problem* (CSP) takes place after *flight scheduling* (determines the flights and their departure and arrival times based on the market demands), *fleet assignment* (assigns the aircraft type to each flight), and *aircraft routing* (individual aircraft is assigned to flights so as to guarantee adequate time for undergoing routine maintenance checks at specific airports)[1]. Purpose of the CSP is to determine individual work plans for all crew members as a sequence of flights and breaks in between, according to various internal and external regulations. However, in operations phase, crews often need to be rescheduled in order to carry out the updated schedule after disruptions. The problem to be solved here is usually called the *airline crew recovery problem* (CRP).

Airline crew management systems typically consist of many sub-systems that tend to solve individual problems related to crew management. Some commonly deployed systems are crew request management, training scheduling, crew pairing, crew assignment, tracking, operations control, etc. These will be discussed in detail later in this thesis.

An airline crew typically receives a monthly or semi-monthly schedule which has to fulfill numerous work rules and regulations. There is a bundle of rigid rules imposed by civil aviation authorities, union contracts, and company policies. In addition, less rigid rules considering crew satisfaction and personal preferences may be applied as well. For these reasons, the problem becomes very difficult to solve, and more complex when the size of the problem increases.

The task of the CSP is to assign all flights of a given timetable together with further activities to a limited number of crew members stationed at one or several home bases. Besides the consideration of all given activities, operational cost has to be minimized, and workload should be evenly distributed among home bases and crew members[2].

Every crew schedule consists of several sequences of flights and other types of activities, assigned to crews in such a way that each flight is covered exactly or at least once by the required *crew complement*. A crew complement consists of a given number of crew members each one belonging to a given position, such as pilot, first officer or cabin attendant. The number of crew members for each position can vary from flight to flight; the crew assignment problem can sometimes be treated separately for each position.

[1] The order described above fits to most American airlines. However, in most European national airlines, the sequence is: fleet assignment, flight scheduling, aircraft routing, and crew scheduling, because of the emphasis of economical use of resources (see Suhl, 1995, for further details).

[2] It is especially the case in most European airlines.

As Fig. 1.1 illustrates, due to its complexity the airline CSP is typically divided into two sequential sub-problems: Firstly, in the airline crew pairing problem (CPP) a set of pairings is generated that minimizes operational cost in such a way that each flight belongs to exactly one pairing. A pairing means a sequence of flights that is carried out as one piece by a given crew ("pair" of crew members). Secondly, the airline crew assignment problem (CAP) or airline crew rostering problem[1] assigns the given pairings to individual crew members taking into account other scheduled activities, such as training, vacation days, and requested off-duty periods. In order to build legal crew schedules for each crew member, an airline must consider all company rules and legal regulations. Therefore, the assignment process may differ from airline to airline, because of different regional or local rules. There are three basic approaches to the airline crew assignment problem: *bidlines*, *personalized rostering* and *preferential bidding*. The traditional crew scheduling approach in North America is based on bidlines, where the set of crew schedules (bidlines) is first generated, crew members place bids on the given schedules, and the assignment is determined by seniority basis. Personalized rostering is usually applied in Europe meaning that individual wishes and restrictions of crew members are taken into account already in the schedule generation phase, and no bidding is needed. Preferential bidding can be seen as a combination of the two first approaches. More details will be discussed in Chapter 3.

1.2 Crew Disruption Management in Airlines

After publication of flight and crew schedules, conducting some slight or major modifications is not unusual for every airline before actual operations. Due to frequent disruptions, such as aircraft mechanical problems, severe weather conditions, sick crews, air congestions etc, schedules are actually seldom operated exactly as planned. Consequently, disturbances to normal operations change the planned schedule to a certain degree, and often require tremendous costs additionally.

[1] In order to differentiate the abbreviations applied in this thesis, CAP stands for both the airline crew assignment problem and the airline crew rostering problem.

Tangible consequences of the lack of operational reliability in airline schedules are flight delays and increasing operating costs due to them. Meanwhile, some intangible losses come from passengers' ill will and time value losses as well (Wu, 2003). It has been reported by the National Air Space (NAS) of the United States that 27% of flights were delayed in 2001. Qantas, the Australian carrier, estimates that 1% improvement of schedule punctuality will bring Qantas an additional $15 million profit in a year.

According to reports of ERA (European Regions Airline Association) (European Regions Airline Association, 2003), most traffic indicators have maintained a steady growth throughout the year. Contrary to the traffic growth, yield has been reduced today to levels last seen in 2000. A considerable percentage of flights has to be rescheduled, even though departure punctuality has shown a steady improvement. During 2003, the percentage of on-time departures, and departures with a delay up to 15 and 60 minutes, was 65%, 86% and 98% respectively. Moreover, within the year 2003 2% of all flights were cancelled, and 34.9% flights were reported being delayed due to various types of disruptions.

Similar observations were also reported in the annual report 2003 of Eurocontrol (European Organization for the Safety of Air Navigation, see Eurocontrol, 2003). The average delay per movement for departures, for all causes of delay, was nine minutes, a decrease of 6.5% within one year. Roughly 40% of all flights were delayed on departure, with 16% out of them delayed by more than fifteen minutes. On the positive side, 11% of all flights departed before their scheduled time. The number of arrival delays fell significantly; down by 7.5% to ten minutes. 38% of the flights were delayed on arrival, with 17% delayed by more than fifteen minutes.

The data above picture the operation environment of an airline literally. They also demonstrate the relatively high frequency of disruptions. It, therefore, turns out to be the reason that an effective *disruption management* plays a crucial role in airlines.

What does disruption management do? Basically, it is a series of actions that an airline takes within disrupted circumstances. The reaction of an airline to any disruption may strongly depend on the type of the disruption, where and when it takes place, what and who are affected either directly or indirectly, and so on. The main concern may be the minimization of customers' (passengers') inconvenience, meanwhile operable subsequent schedules have to be carried out within a short period of time.

Because of disruptions, parts of crew schedules become no longer feasible. For instance, the originally scheduled aircraft has been rerouted, which may require the substitution of crews because of a change in aircraft type (originally assigned crews may not be qualified to operate the new plane). In such a case, crews who are available may be called in to serve the flight. Further rescheduling tasks may be necessary, in case there are not enough crews available.

Therefore, *crew recovery* aims to find a solution which includes the rescheduling of crews so that any changes caused by disruptions are considered. It is to find the "right" people to operate the "right" flights at the "right" time. Every flight has to be properly served by a number of required crew members, so that the airline does not need to pay too much extra cost.

1.3 Decision Support in Airline Crew Management

Generally speaking, decision support systems (DSS) are computer based systems which support managers, planners or controllers in core decision functions in all divisions of an enterprise. Since the introduction of DSS in the 1970s, they have received great attention which has led important development activities over decades. Instead of replacing decision makers, a DSS is meant to be an adjunct to key decision makers to extend their capabilities and thus to support and improve the efficiency and efficacy of decision making. A DSS may solve or assist in solving considerably complex problems by applying techniques developed in areas of Operations Research (OR, also known as Operational Research) and Management Science (MS). OR/MS is traditionally characterized through the use of mathematical techniques and models to support decision making which are able to cope with the complexity of airline crew scheduling and recovery. For example, the two key components in OR/MS, *optimization* and *simulation*, have been systemically studied, and appear in many decision support systems.

Nowadays, the capability of a decision support system has been greatly enriched by combining more and more new emerging technologies. The substantial progress of DSS during recent years comes from the significant improvement in algorithms, problem solving methodologies, software development, hardware, problem process development, knowledge management, etc., together. Today decision support systems are more and more used in key decision making processes, assisting users to make crucial decisions with highly complex and dynamic characteristics. According to the fast growth of hardware technology and operations research methodologies, decision support systems of today are able to handle problems that need mass computation for finding optimal solutions, which was not realistic some years ago.

Within tourism airline industry, various decision support systems have been widely applied since many years to support solving complex problems encountered by airlines. However, the boom of the air transport sector and the expansion of the network coverage lead to a wider range of difficulties than anytime before. Because of such a fast growth of the airline industry, there are urgent needs of further substantial support, which spur the scientific efforts.

Traditionally, DSS play an important role in the airline schedule planning process, whereby aircraft and crew scheduling are most important and complex planning problems. The general task is to manage scarce resources efficiently and effectively in order to meet the public transportation demands. Because of a large number of aircraft, crews and flights, producing schedules may require days or weeks of work, if it is carried out manually by humans. Furthermore, resources involved in these processes are usually rather expensive, so that every decision is actually cost intensive. Therefore, high cost savings can be achieved through systems to support crucial decisions of the schedule planning.

Because of the complexity of the airline crew scheduling problem involving a huge number of crews and flights, great attention has been paid to it by many researchers over the years. Many scientific publications within the last years show rapid development in introducing efficient algorithms and building comprehensive decision support systems for airline crew scheduling.

However, the problem of rescheduling crews becomes more and more crucial recently. Disruptions happen frequently, constantly affecting the normal operation and introducing chaos, but there is a lack of dedicated methods and systems for airlines to recover their operations. For most airlines, the recovery is primarily a manually driven decision process, which involves complex decisions that cannot be easily handled by humans manually. Especially, the recovery of crew schedules compared to the recovery of aircraft is very sensitive because the resource involved is humans instead of machines. Such a particular need motivates this research to design and develop a decision support framework for solving the airline crew recovery problem, in which many different techniques and strategies can be combined.

1.4 Organization of the Thesis

This thesis is organized as follows. In Chapter 2 the detailed description and definition of the airline crew recovery problem is given. It is followed by the literature review (Chapter 3) in which state-of-the-art techniques related are presented regarding both airline crew scheduling and rescheduling problems. In Chapter 4, the problem is mathematically formulated, and corresponding exact optimization methods are presented. Heuristic solution methods are presented in Chapter 5, which includes a genetic algorithm based method and a constructive algorithm. In order to apply proper strategies to solve such a problem, a classification of possible disruptions and a corresponding strategy mapping are introduced in Chapter 6. In Chapter 7, a dedicated decision support system and its major components are described in detail. Finally, in Chapter 8 conclusions are made based on the results achieved, and the direction of the future research is given in the end of the thesis.

Chapter 2

The Airline Crew Recovery Problem

Anecdotal evidence suggests that most airline carriers never experience a single day without disruptions. Planned operations are often changed based on various types of disturbances. In the setting of passenger airlines, a *disruption* is a situation in which an airline is prevented from normal operations as planned because one or more unexpected events happen. Most disruptions, as disturbances to airlines' normal operations, have dramatic impacts in many ways. Within a disrupted situation, passengers may get stuck at airports because of cancellations or delays of their flights, which definitely makes them dissatisfied with the services provided. An airline may face a temporary shortage of flight crews or aircraft, which makes it more difficult to recover and operate later flights. Furthermore, disruptions that occur simultaneously or closely to each other may interfere and imply even more serious problems if they are not managed in a proper way. During irregular operations, the operations control center (OCC) of an airline is usually the department in charge to handle all disruptions that occur.

As described in Chapter 1, expenses paid for the management of airline crews are extremely high, especially for those highly skilled airline crews who operate aircraft, so that effective management of flight crews implies great cost reduction. If operations are disrupted, a large amount of money has to be paid in order to get back to the original schedule. For example, more aircraft may be needed, reserve crews may be called in, compensation to passengers may be paid because of flight cancellations and so on.

To diminish the impact of disruptions that cause serious problems, an airline has to do many things. Basically there are two fundamental ways that can help airlines to reduce disruptions significantly. The first way is to establish "robust" flight and crew schedules ahead of their actual operations. The term *robust* or *robustness* of a schedule indicates that the schedule published cannot be easily affected by certain types of disruptions and can be degraded locally with the minimal impact on the entire schedule. As an example, the degree of the robustness of a schedule can be achieved to a certain extent by relaxing the durations of those short layovers that are very likely to be disrupted and cannot be easily recovered. The issue of establishing robust crew schedules is not the focus of this work, we, therefore, refer to Ageeva (2000), Chebalov and Klabjan (2002) and Klabjan and Schwan (2000). The second common way to do so is to deploy a recovery system that can bring back normal operations in a proactive manner quickly and efficiently. As the focus of this work, details of a crew recovery system will be described in the later chapter.

This chapter starts with a brief description of the operation environment of airlines and the general crew recovery problem are described briefly in Section 2.1. It is followed by Section 2.2, in which the detailed structure of such a problem, including the resources involved, activities, constraints, disruption scenarios, disrupted/recovery period and cost structure, are discussed in detail. Section 2.3 addresses the general objectives of crew recovery problem. Furthermore, a brief review of airline crew recovery processes in practice is given in Section 2.4. Finally, Section 2.5 gives a short description of the testing instances examined in the research.

2.1 Problem Environment

In this section, we will give a brief overview of the operation environment of the airline crew management. It is divided into two subsections: In Section 2.1.1 we will present the basic planning process that an airline usually carries out to generate schedules; in Section 2.1.2 we will elaborate on the airline's operation recovery process in disrupted situations.

2.1.1 The Planning Process as Basis for Operations

In most major commercial airlines, e.g., U.S. domestic operations, a *hub-and-spoke network* is often applied. Within such a network, each *hub* represents a high rate of departure/arrival of flights, while a *spoke* is an airport with a limited amount of daily departures/arrivals connected with one hub. In contrast, some European airlines and most international flight networks adopt so called *point-to-point networks*, in which flights are operated between pairs of airports. Due to the significant difference between hub-and-spoke and point-to-point operations, we may need different models and approaches for both. In this thesis, we mainly focus on the latter taking into account special features of European tourist airlines. Such special features have been more or less neglected in scientific literature, whereas there are many publications available focusing on hub-and-spoke networks.

Within the airline schedule planning process, several sub-processes, namely flight scheduling, fleet assignment and aircraft routing, must be finished before crew scheduling actually starts (Antes, 1997). At the beginning of crew scheduling, each *flight leg* (also called *leg* in short, meaning a non-stop flight trip from one airport to another) already has a fixed departure/arrival time and an associated aircraft type (for example Boeing 737-300, Boeing 747-400, Airbus 310-300, Airbus 300-600 etc). With regard to the given flight service demand, the crew scheduling process partitions flight legs into a hierarchical set of sequences: *flight duty*, *pairing*, and *roster* (also called *line-of-work, LoW*). Flight duty, equivalent to *duty period* or simply *duty*, is a set of consecutive flight legs which can be legally assigned to one single crew member. Normally it refers to one day's work of a crew member, satisfying all required rules and contractual restrictions. The duration of a flight duty normally starts 1 hour before the departure of the first flight on duty (*briefing*) and ends 15 minutes after the arrival of the last flight (*debriefing*). A pairing normally consists of one or more flight duties, which starts and ends at the same airport (called *home base*) where crews usually start their service, while a roster is the schedule of a crew member within the *planning period* given (e.g., a complete half or one month work schedule of a crew member can be considered as his or her individual roster).

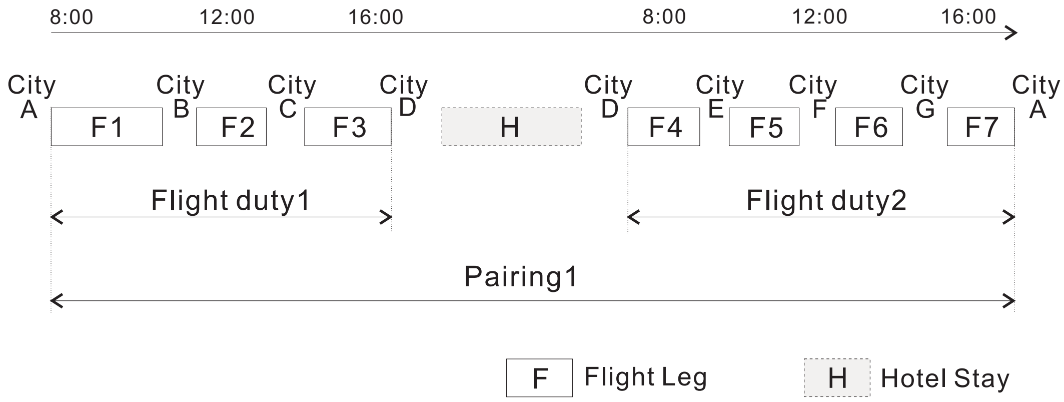


Figure 2.1: An example of crew pairing

An example of crew pairing is given in Fig. 2.1. It includes two flight duties and one overnight hotel stay at airport D, where airports with possible hotel stays are called *hotel bases*. An overnight stay (e.g., H depicted in Fig. 2.1) at city D is necessary for this crew member due to the fact that his/her first flight on the next day (flight F4) starts from the same city. By chaining several pairings together during planning, airline planners can form one possible roster for a specific crew member, which satisfies all the rules that are relevant to such a process.

As shown in Fig. 2.2, besides typical flight services an airline crew member is also involved in regular training events, flight simulator and office work, and so on. A limited number of vacation days are also guaranteed with respect to the regulation imposed by civil aviation authorities, labor unions, and the airlines themselves. Together with off-duty days requested by crews (called *requested-off*), the three kinds of activities mentioned above sketch the availability of crew members which in turn represent the crew capacity within the given planning period. With the availability information of every crew member, planners are able to create personalized schedules for everyone prior to the actual operation of flights.

Usually the crew scheduling task in the planning phase can be performed in either a sequential or an integrated fashion (see Fig. 2.2). Traditionally, common research adopts the sequential approach by dividing it into two sub-steps: crew pairing and crew assignment. However, some integrated approaches have also

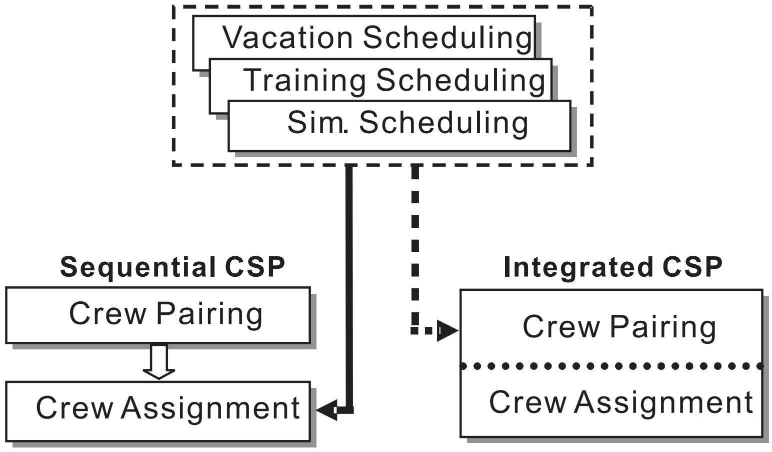


Figure 2.2: Airline crew scheduling process

appeared recently, and will continue to be the future direction due to the fast growth of computing power. Some more details will be given in Chapter 3.

Crew scheduling requires an optimally scheduled coverage of all flights with regard to given flight timetables. Large airlines usually use computer-based optimization techniques to determine a cost-minimal crew schedule. Depending on the size of the instance, each sub-step may require a long computational time to find an optimal solution, ranging from minutes to even days or they may be non-tractable with optimization methods. For extremely large instances, the actual goal is, therefore, to find a solution close to an optimal one by applying heuristic techniques step by step. More details regarding the solution methods of the airline crew scheduling are given in Section 3.1 of the next chapter.

In most research, the topic of airline crew management, especially the topic of airline crew scheduling, focuses on *onboard crews* (also called *flight personnel*, *flight crews* or *aircrews*) including two groups: *cockpit crews* and *cabin crews* (also called *flight attendants*). The crews work in the cockpit and cabin, respectively, to operate the plane and to provide service to passengers. Depending on the type of the aircraft, a flight leg is assigned to a certain crew complement with given crew positions and a given number of crew members per position required for the flight. There are significant differences between the scheduling processes for cockpit and cabin crews, because of different legal regulations and union agreements, as well as the different group size per aircraft, so that optimization models for cockpit and cabin should be developed and solved separately. In this work, we focus on the requirements for cockpit personnel, because they build the more expensive crew part with more complex regulations. The methods developed for cockpit crews can be then applied for cabin personnel as well. Hence the term crew intends to mean cockpit crew in the rest of the thesis unless there is a particular explanation. Regarding the scheduling problem for airline cabin crews, we refer to Day and Ryan (1997) and Kwok and Wu (1996) for more details.

2.1.2 The Recovery Process at Operations Time

Although flight schedules have been published, actual operation of a schedule is subject to many internal and external factors which may induce changes to the schedule. A schedule may thus be modified in scenarios caused by disruptions. In reality, it is often a fact that frequent disruptions imply high additional costs in today's complex and uncertain operational environment, namely schedules are seldom operated exactly as planned. On the contrary, they are constantly disrupted by irregular events during day-to-day operations. As a result, disturbances to normal operations change the planned schedule totally or at least partly. More importantly, tremendous costs have to be paid in order to recover from them.

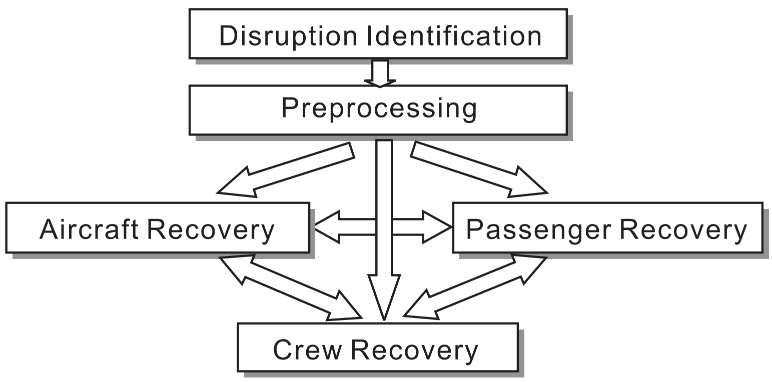


Figure 2.3: Airline crew recovery process

The crew recovery process takes care of disrupted situations in which original crew schedules require several, sometimes major, modifications to keep the airline's operations running after an unplanned occurrence. When disruptions happen, a set of flights has to be delayed and even cancelled. Additional aircraft, crews and flights are required in order to have enough resources to serve all the flights that need to be operated. Usually, aircraft are first rerouted to cover disrupted flights, and to pass maintenance airports for regular checking. In this process the flight schedule is modified. The rescheduling of crews is then carried out based on the newly updated flight schedule and crews' availabilities. After the generation of new crew schedules, a certain number of crew members influenced by the updated schedule are notified through the communication system.

Fig. 2.3 illustrates the basic steps that an airline takes to recover its disrupted schedules. Disruptions can be identified either by an automated system or manually. However, some disruptions cannot be easily identified, e.g., in a case that a given delay causes further delays for subsequent flights in completely other parts of the schedule. Sometimes few short delays may not cause any problem because there is a way to absorb them by pre-scheduled buffer time between flights. However, in other cases even a minor delay may cause severe problems, if it cannot be compensated through buffer time. In the case of severe weather conditions, the disruption information must often be collected manually. Clear and sufficient information regarding the disruption has to be gathered, such as its source and duration, who or what is affected, and so on. Furthermore, a "snapshot" of the current situation has to be presented to the decision makers: status of each aircraft, location of crews, situation of each affected airport, real-time information of every flight, etc.

The airline crew scheduling process has basically to reschedule a subset of the flights that the crew scheduling process has taken care of. In the recovery case, those flights that are directly or indirectly disrupted by irregular events have to be reassigned. Besides those crews that are in operation, two additional groups of crews may be considered: *standby crew* and *reserve crew*. Standby crews are crew members positioned at large airports (normally home bases), ready to substitute any other crew member who is not able to fly her/his flights. Reserve crews normally stay at home being ready to be called to serve open flights that cannot be assigned to any other person. When calling reserve crews, a predefined period of time is given to allow them to get to the airport, and be ready to fly. If there are not enough crews available at an individual airport, crews from other airports may be transferred by taking a plane (*deadhead*), or by another public transport system (*transit*, e.g., by taxi, train, etc).

Table 2.1: Comparison between CSP and CRP

	Crew scheduling	Crew recovery
Activities	.Scheduled flight legs .Pre-scheduled activities .Requested-off	.Scheduled flight legs .Pre-scheduled activities .Requested-off .Updated flight legs .Newly scheduled flight legs
Crew	.Operating crew	.Operating crew .Standby crew .Reserve crew
Duration	.One month or half a month	.Hours .Days
Time	.Weeks ahead of operations .Revise few days before operations	.Daily .Directly after disruption(s)
Cost	.Transit cost .Hotel cost	.Transit cost .Hotel cost .Cost of using standby crews .Cost of using reserve crews .Cost of changes

Table 2.1 shows the comparison between the two processes at different stages: planning and operational phase. The comparison is made in terms of the activities involved: crews, duration, costs and times when they take place. As one can see from the table, the primary differences that the recovery process possesses can be explained as follows:

- More activities are involved, e.g., updated and newly added flights
- Standby and reserve crews are considered additionally
- The time horizon for the recovery process is much shorter than the one in the scheduling process
- Times when the two processes take place are different

- More cost factors are involved in the recovery process

A more elaborate description is given in the next two sections.

2.2 The Structure of the Recovery Problem

In this section, the general structure of the airline crew recovery problem is given by introducing the resources involved, classification of activities, constraints, disruption scenarios, the disrupted/recovery period and the relevant cost structure.

2.2.1 Resources Involved

Within a disrupted time period, three kinds of resources must be recovered: aircraft, crews, and passengers (see Fig. 2.4). Each resource has a great impact on the new schedule. For example, a shortage of aircraft may cause not only unexpected delays and cancellations, but also some additional difficulties to the later crew rescheduling, because crews may lose their connections or get stuck at an unfavorable airport. Due to the complexity, the overall recovery problem is usually decomposed into a sequence of sub-problems, each of which is solved independently. Usually, the aircraft recovery problem is solved first so as to restore the flight schedule with respect to all company rules and maintenance requirements. The impact of disruptions upon passengers is reduced as much as possible by minimizing their inconvenience, such as missing connections and further delays. Finally, crews have to be rescheduled under the updated situation. Notably, the way to decompose the entire recovery problem differs from airline to airline because of heterogeneous company rules. The reason for applying a sequential approach relies on the fact that a completely integrated three-phase problem cannot be solved with today's technologies because of its extremely high complexity.

Basically, all resources involved in a disturbance have to be reconsidered or reallocated. This makes the overall recovery problem extremely difficult to solve, as each single sub-problem might already be a rather complex task. Moreover, any changes regarding one resource may have a distinctive impact on the total situation, which, in turn, may cause further conflicts.

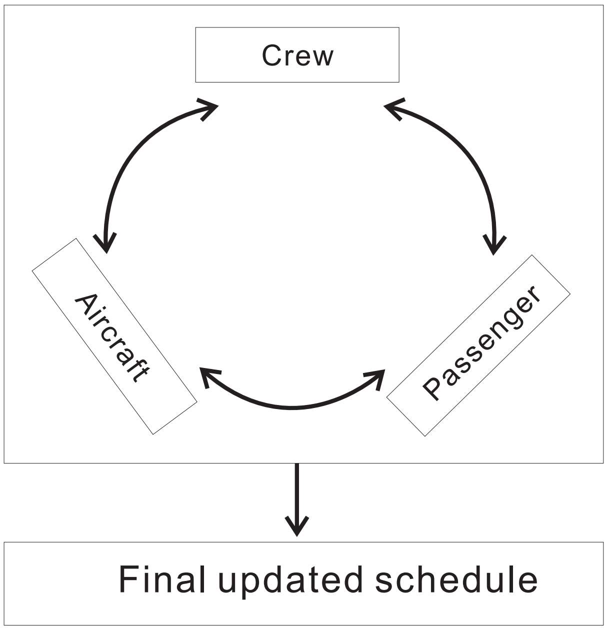


Figure 2.4: Resources involved in the recovery process

Since the purpose of this thesis is to study the airline crew recovery problem, we assume that the aircraft recovery problem has already been solved prior to the crew recovery. In other words, a newly proposed flight schedule is already given where some flights may be cancelled, delayed, rerouted or added. We define all these flights as *affected flights* through the given disruption. Thus the departure or arrival times and types of the aircraft of affected flights are known before airlines reschedule their crews.

2.2.2 Activities of Cockpit Crews

Basically, there are three groups of activities: flights, pre-scheduled activities and requested-offs. Generally speaking, all three groups have to be included in the final updated crew schedule, i.e., assigned to individual crew members. However, crew requests may not be satisfied in some situations where more crews are needed to cover all open flights.

In the group of flights within the affected time period, originally scheduled flights normally constitute the largest portion of all flights. Exceptions are given in extreme situations, such as the September 11 terrorist attack on the World Trade Center in New York City. The rest of an updated schedule includes rescheduled flights (with updated departure or arrival time) and newly added flights. New flights are usually added when a flight has been cancelled for some reason.

Pre-scheduled activities are those activities assigned to crews as parts of their daily work, so that their starting times and durations are predefined. Typical pre-scheduled activities are:

Vacation: for example yearly vacation.

Simulator: a special training with flight simulator of a certain type of aircraft required to maintain a crew's qualification.

Office duty: regular office work

Courses: other courses or training besides simulator provided and required by airlines

Other: further ground duties, such as medical checks

2.2.3 Constraints

When solving the crew recovery problem, a bundle of rigid rules and regulations have to be applied which are imposed by civil aviation authorities in each country, union contracts and company policies. For example, in the United States the Federal Aviation Authority (FAA) regulates airlines' operation for crews' safety. Such regulations limit the length of duty periods and specify the rest necessary between duty periods. In Germany, the Luftfahrt-Bundesamt (LBA) is fully responsible for similar regulations concerning every crew scheduling (and rescheduling) problem.

Although the structure of the problem remains the same for all major North American airlines, specific collective agreements change this picture slightly. In Europe, collective agreements are usually much stronger than governmental regulations. Collective agreements are typically very detailed and change frequently. In addition, some rules that apply locally are quite different from airline to airline.

Rules can be seen as "hard" or "soft" with respect to their rigidity. Hard rules often imposed by civil aviation authorities and union contracts, have to be followed by an airline without any violations. Soft rules, in contrast, may be violated in some specific situations, which have been clearly documented. A schedule is considered to be *legal* only when it fulfills all relevant hard and soft regulations and rules.

As described in Section 2.1, the flight schedule (roster) for one crew member can be decomposed into subsequent levels: pairings, flight duties and flight legs. There are a large number of rules that have to be applied on each roster, flight duty and flight leg. Typically, they express restrictions on the length of the working periods as well as require appropriate rest periods between flight legs, flight duties, and pairings. In the following, the most important rules are briefly described:

- Maximum daily/weekly/monthly/yearly flight hours, for example, the maximum flight time between two daily rest periods is limited to at least 10 hours, and it can only be violated by extending to up to 14 hours if the crew member gets an extra off-day next day. Maximum monthly or yearly flight hours normally are determined through the individual contract of a crew.

- Maximum flight duty hours, normally referring to the hours without off-duty rest time.

- Minimum off-duty interval, meaning the required interval between two consecutive flight duties.

- Maximum time away from crew's home base: It restricts the total time that a crew member can work outside his or her home base. Certain compensation policy is applied in this respect by every airline.

- Minimum daily/monthly flight hours: A certain number of flight hours are required based on an individual crew's contract. Usually, in European Airlines the planners try to distribute extra work evenly among crew members.

- Maximum number of daily landings: For example, the total number of landings that a crew has within a day cannot be more than 4.
- Minimum daily/weekly/yearly rest: For example, between any two flight duties, a 10-hour minimum rest is guaranteed. It may increase if the flight hour served exceeds 10 hours. Another commonly observed rule is that two consecutive off-duty days must be given within each seven day period.

Moreover, other restrictions may be also applied on overall crew schedules. For example, an upper bound on the complete cost of the final solution may be given, pre-calculated based on instances. In some airlines, especially in European ones, flight hours should be evenly distributed among home bases and crew members. This procedure guarantees a certain degree of fairness. In the case of crew recovery, one particular rule is set to reduce the completing time of the recovery, e.g., minutes instead of hours is fairly desirable. However, this cannot be explicitly modeled, but by applying particular algorithms or methods that may reduce the complexity of the problem examined. This issue will be discussed in more detail in Chapter 6 and 7.

2.2.4 Disruption Scenarios

Generally, the number of different causes of airline disruptions is large. They can be grouped into the following seven categories:

Weather: Flight traffic is very sensitive to weather. Inclement weather conditions, such as heavy snow, storm, typhoon, and so on, often cause severe disruptions affecting a large number of flights and crews.

Flight operations: Disruptions may be caused through an airline's own procedure (or those of its handling agent), especially during the period of pre-flight preparation, such as loading, aircraft landing etc.

Aircraft/equipment technical problems: Disruptions may be caused by technical problems of an aircraft or problems of equipment on board, such as aircraft engine failure, communication system problem etc.

ATC (Air Traffic Control): ATC often assigns delays induced by the general air traffic situation, e.g., a requested departure 'slot' of a given flight may not be available.

Reactionary: This category means late arrivals of incoming aircrafts, causing further delays. Such a delay is commonly observed as a primary delay inducing a disrupted schedule. For example, an aircraft which suffers an ATC delay at the start of its working day may carry that through as a reactionary delay on subsequent flights until the delay is absorbed.

Passenger: Individual passengers may cause disruptions through late boarding or other specific reasons.

Other: There are a number of further possible causes that do not belong to any categories listed above but affect the normal operation.

Generally, reactionary causes constitute the most frequent disruptions to normal operations, followed by delays caused by operations of ATC. As reported both in European Regions Airline Association (2003) and in Eurocontrol (2003), around 35% of delay causes are reactionary delays, which are propagated over time.

In order to develop a decision support system to assist disruption management, we would like to classify disruptions and the techniques to cope with different types. One possibility is to classify the disruptions according to their severeness into three groups: *minor*, *medium* and *major*. A disrupted situation is treated as a minor disruption when there are only few affected flights (those flights that have to change their schedule may be reassigned to other crew members). Major disruptions, however, involve a huge number of affected flights. For example, a major disruption occurs when a busy hub experiences a serious snow storm which causes stop of service for hours. The rest of disruptions that lie in between minor and major groups belong to medium scenarios.

However, there are no clear boundaries for the three groups of disruptions introduced above. This is due to the fact that a small disruption, such as delay of a single important flight, may imply a large number of changes, and sometimes a seemingly large disruption may be handled with only a few changes. On top of that, the seriousness of a disruption may be measured in several ways. The following criteria show some hints of measuring the impact of disruptions on the current operation:

- The total number of flights in the instance examined
- The number of flights that are directly affected by disruptions
- The number of flights that are affected due to the propagation of delays and cancellations
- The total number of crew members in the instance examined
- The number of crew members who are affected by disruptions
- The number of standby and reserve crew who are available during the period, especially those who are available at the airport where disruptions occur
- The number of daily flights in average
- The total duration of the time period with changes in the schedule

Each factor alone is not able to represent the seriousness of a disruption. Instead, they should be considered as a group when one tries to investigate the disruptions occurred. In the following, we discuss an example that shows how to measure the seriousness of a disruption in a straightforward way (notably, this may not be applicable for some cases). We assume that during a pre-investigation a set of flights has been rescheduled either in terms of their departure/arrival times or through changing their airports. We consider a situation with less than 5% of total flights changed as a 'minor' disruption, and a situation with more than 15% of all flights that has been rescheduled in order to recover from the disruption as a 'major' disruption. The rest in between can be seen as 'medium' disruptions which have moderate impacts on the current operation.

In order to handle the problem more efficiently, airlines have to make concerted efforts to develop specialized strategies for each specific case. The strategy adopted, therefore, must reflect the seriousness and characteristics of the given disruption. Further relevant discussions are given in Chapter 6.

2.2.5 Disrupted and Recovery Period

After rerouting aircraft and considering passengers' connections, some flights are delayed, cancelled or newly added. That also can be understood as a set of flights whose departure or arrival times have been updated. Eventually, the earliest and latest updated flights can both be easily found out.

Before starting the actual recovery process, coordinators in an operations control center usually have to make efforts to ascertain the range within which flights can be possibly reassigned to different crews. Outside such a range, no flights are allowed to be reassigned. The purpose of this is to reduce the difficulty induced by the increasing number of activities involved. A longer range of flights may produce many more possibilities to find a "good" solution. Meanwhile, this can also be interpreted as more open flights that need to be reassigned, which normally requires intense computations.

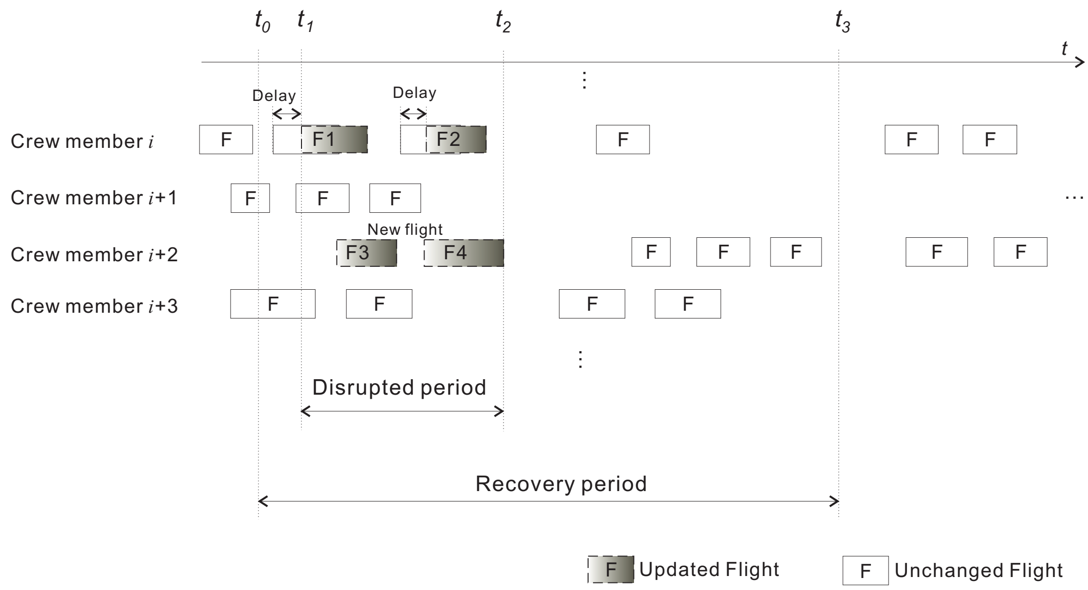


Figure 2.5: Disrupted and recovery periods

In our approach we distinguish between *disrupted period* and *recovery period*. The disrupted period starts from the departure time of the first updated flight (the schedule of the flight that has been changed), and ends with the new arrival time of the last updated flight. As illustrated in Fig. 2.5, the period between t_1

and t_2 is defined as the disrupted period, where t_1 is the new departure time of flight leg F1 and t_2 is the new arrival time of F4. Further details will be discussed in 4.1.2

In contrast, the length of the recovery period is not deterministic. It denotes the period to recover from a disruption, with other words the recovery period lasts until all changes caused by the disruption have been carried out, where there may be many more changes than the originally disrupted schedule. The recovery period starts at the same time with the beginning of the corresponding disrupted period or earlier, and ends at the same time with the end of the disrupted period or later. The decision of the length of such a period strongly depends on the scale of disruptions and the size of the instance. For example, one may choose a considerably longer recovery period, if the size of the instance is not large and the effect of disruptions is not significant. Likewise, a shorter recovery period may be helpful in recovering from a large instance, simply because only few activities and crews are considered.

2.2.6 Cost Structure

The cost of airline operations recovery can be excessively high. Taking aircraft as an example, an airline has to reroute some aircraft in order to carry out flights in disrupted situations, which may require more fuel. Extra aircraft may be ferried in urgent situations, which often costs a large amount of money. In the rest of this section, we solely discuss the underlying cost structure within an airline crew recovery process.

Regardless of the difference between the crew scheduling in the planning phase and the crew recovery in the operational phase, the large proportion of cost for both processes are operational cost and crew payment. The operational cost is the sum of the cost of all assignments which require extra money to make schedules feasible. Further description is given in Section 2.3.1 and 4.2.

The payment for airline flight crews may vary depending on airlines. For example, in the U.S. airlines do not measure the cost of a crew schedule in monetary terms. It is calculated in terms of minutes of *pay-and-credit*, and crews are paid proportionally to the number of pay-and-credit minutes they accumulate (further details we refer to Barnhart et al., 1999a; Gershkoff, 1989). In contrast, in Europe the payment for crews is normally based on a regular salary for up to a certain number of flight hours per month or year. The monthly base payment is applied independent of the number of flight hours a crew member flies monthly. The excessive flight hours have to be paid additionally. Moreover, further cost has to be paid for calling in any reserve or standby crew members during the recovery process, normally in the situation of "tough" disruptions. The number of reserve and standby crew members who are called to work should be minimized because of the high cost that has to be additionally paid. The fact that a number of reserve and standby crews have to remain available for any possible further disruptions causes higher cost even if the persons are not called for duty.

2.3 General Problem Objectives

Factors that determine economical sustainability of passenger airlines include cost efficiency, good yield management, service quality, appropriate network coverage, and so on. Because each of these factors has a dramatic influence on the overall performance of an airline, working on one or more of them may partially compensate for the high cost. Saving of cost becomes extremely important in cases with a fast increasing cost component, such as high fuel price today. Consequently, this leads to a high complexity of strategies.

In the case of airline crew recovery, various objectives have to be considered, including some that are particularly important for this type of problem. Generally, cost efficiency is not the only determining objective for the crew recovery process. Service quality, satisfaction of crew, appropriate recovery period, and problem solving time are also highly required from airlines and their customers.

In this section, three objectives will be examined and described in Section 2.3.1, 2.3.2, and 2.3.3 respectively.

2.3.1 Minimization of Additional Cost

The major costs concerned in the crew recovery are those additional costs paid to reschedule crews. In order to serve all open flights, standby and reserve crews may be called to work, which imposes additional payment to them. Furthermore, a large amount of money has to be paid to transfer crew members from one airport to another to serve their next scheduled flight. In every airline, there are formulae which are adopted to calculate it. Moreover, costs for overnight hotel stays (normally at airports with pre-selected hotels) are also calculated.

Therefore, the most concerned objective is to minimize the sum of the three groups of costs incurred by additional new schedules. Usually such extra cost of the recovery can be reduced by finding a solution with less hotel stays, transit and also fewer standby/reserve crews who are used for substitutions. For further details, we refer to 4.1.1.

2.3.2 Solution Time Restriction

In the crew recovery problem, the time restriction becomes more serious, compared with the crew scheduling in the planning phase or problems in other sectors. Unlike the planning process, airlines immediately need a recovery solution whenever they find it necessary to reschedule because of disruptions. It is usually the wish to bring back the schedule planned within minimal period of time.

Often it is desirable to find a solution of a recovery problem within minutes, since longer time may cause further propagated disruptions over time. In some situations, however, a longer solution time may be necessary in order to find at least one feasible recovered crew schedule due to the complexity of the problem. Nevertheless, airlines cannot afford to wait for hours for an optimally recovered solution. Today in practice, usually a feasible, not optimal, solution has to be chosen, even though tremendous cost has to be paid.

2.3.3 Crew Disturbance Reduction

Another objective is to reduce the disturbance to all crews who have been affected by disruptions and the final updated schedule. This can often be measured by crews themselves. One simple way is to evaluate if crews are willing to accept their new schedule, and if there are difficulties for them to change schedules. In short, general criteria are needed to measure the disturbance of crews, in terms of convenience, fairness etc.

Crew convenience can be increased either by minimizing the number changes scheduled for the following days, minimizing the total number of notifications about changes over all crew members, or simply minimizing the number of crew members that face changes in their schedules. Reducing the number of notifications expresses that fewer crew members will be informed about new schedules. Therefore, fewer crews may possibly complain about the updated schedules. Furthermore, savings in communication costs can be achieved as well.

2.4 Crew Recovery in Practice

As described previously in Section 2.2.1, three resources are involved in the overall recovery process: aircraft, passengers and crews. In practice, the three resources are controlled separately by different sub-departments of an operations control center. As illustrated in Fig. 2.6, *crew coordinators*, *aircraft coordinators* and *passenger coordinators* are responsible for each disrupted situation, and provide proposals of relocations of individual resources.

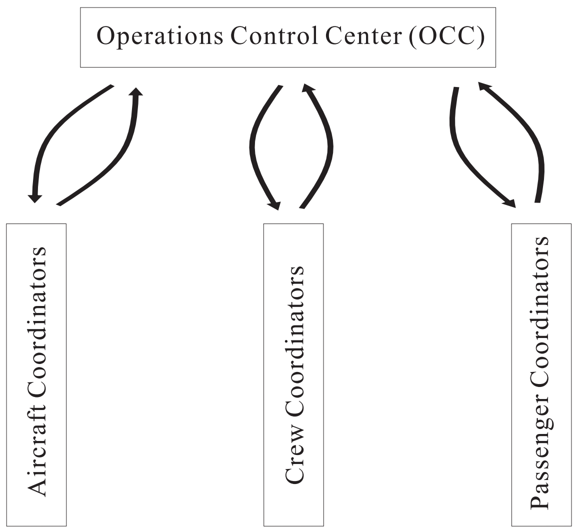


Figure 2.6: Operations recovery

The author of this thesis together with a graduate student recently conducted an interview with operations control personnel in one of the major European airlines[1], especially with operations managers and crew coordinators. The main purpose of the interview was to conduct an analysis regarding practical issues of the airline crew recovery problem.

As the airline reported, they usually divide disruptions into two categories: crew-triggered disruptions and operations-triggered disruptions. Crew-triggered disruptions, roughly 25% of total disruptions, are those problems caused by crews themselves directly or by a factor related to them. Possible examples include:

- Violations of rules, regulations and contractual limits, e.g., duty hours, rest time etc.
- Missing luggage of a crew member
- Illness of a crew member
- Qualification problems that happen when a crew member is not qualified to operate or serve a certain type of aircraft
- Missing crew members

If crew-triggered disruptions happen, they are usually forwarded immediately to crew coordinators, skipping to report to other departments in the OCC in order to speed up the recovery process.

By operations-triggered disruptions, we mean disruptions caused by all types of operational irregularities, such as flight cancellations, non-regular maintenances, bad weather, rescheduled flights, etc. During the interview, we were told that they constitute the majority of disruptions faced by airlines, 70% in general. Normally, when such a disruption happens, the problem is first reported to the OCC where it is measured and classified preliminarily. The pre-examined problem, therefore, is forwarded to one or more divisions depending on the type of the problem. For instance, a disruption that causes not only rerouting aircraft but also rescheduling crews has to be handled by both divisions simultaneously. Further interactions often take place between OCC and its three departments. The final confirmation is given by the center before any proposed rescheduling is

[1]The interview was carried out at Lufthansa Passage, Frankfurt/Main

carried out, due to the fact that any decision may have an impact on the recovery in other departments.

In the crew control center, crew coordinators try to find a feasible recovery solution by using a system that provides a graphical representation of crew schedules. Certain actions may be taken to build a feasible and "good" recovery solution. Most common strategies are:

- Using stranded crew
- Crew swapping
- Deadheading or transferring crew
- Using standby/reserve crew
- Delaying flights for a short time, e.g., minutes
- Delaying flights for a long time, e.g., 1 hour
- Advancing flights for a short time
- Advancing flights for a long time
- Cancelling flights

The decision of choosing a specific action is subject to various factors: (1) Whether it is expensive; (2) Whether it can cause further changes to aircraft routes or difficulty to passengers; (3) Whether it can be applied easily. However, some general principles are applied as rule of thumb. For instance, using stranded crew members is always desirable because the airline does not need to pay additionally. Standby or reserve crews are also considered as an expensive resource, hence the use of them should be minimized. Notably, delaying or cancelling flights are usually treated as the last resort since it may produce further impact in many other sectors.

In solving the crew problem of irregular operations, the most pressing issue is often not to adopt the optimal solution that has normally been determined during the crew scheduling in the planning phase. Instead it attempts to find solutions with "good" quality within the time frame faced by crew coordinators. It appears to be not reasonable if no solutions are taken in the end of the recovery, while it is possible for planners to re-optimize in the crew scheduling stage. During the interview, we were told that the crew recovery is usually completed within 45 minutes manually, regardless of the type of disruptions (as long as the disruption is controllable for the airline).

As has been reported in Wei et al. (1997), in most major airlines in the U.S., the task can be addressed as: (1) to find crews for disrupted flights whose crews are not available due to disruption, and (2) to fix the broken pairings caused by the disruption. In other words, it is to cover as many disrupted flights as possible. Therefore, the problem can be stated either as covering all flights while maintaining the integrity of a maximal number of crew pairings or as repairing the disrupted pairings while covering a maximal number of flights. In practice, crew coordinators follow the so called "buy-time" strategy. This makes them focus primarily on the current moment, and solve the most urgent problem first. In some cases they may even solve the current problem by creating a new problem that will be handled later.

Currently, the crew recovery is primarily a manually driven decision process, i.e., decisions are usually made in an empirical manner based on limited information and support. Only few airlines report that they use decision support systems to deal with this particular problem (sample applications can be found in Section 3.2).

2.5 Test Instances

All test instances presented in this thesis are from a medium-sized European tourist airline, where its operating network is a mixture of a hub-and-spoke network and a point-to-point network. Within such a network, multiple home bases are located in Germany, while many other airports are spread out around Europe. The airports outside Germany are normally resorts which attract a large number of travelers every year. Passengers, therefore, usually spend some days at a destination and come back a few days or weeks later. An effect of such a characteristic is that a large portion of the flights can be organized as round trips. In other words, a typical schedule for one crew member may be a trip to one city, returning within the same day if he/she has only two flights to serve. Sometimes a crew member may not fly back directly after he/she arrives at the destination, but goes somewhere else instead. This happens when his/her next flight starts from that airport heading for another place. However, it is also usual that there are other flights going back to his/her departure airport, and the flight is operated by other crew members.

In the setting of the airline involved, crew members, particularly cockpit crew, are qualified to operate only limited types of aircraft. Therefore, airlines group their aircraft into fleets regarding an aircraft's generic specification. This makes it possible for us to decompose the problem and examine it fleet by fleet. For example, we use 'A' as the first letter of one fleet initially, which denotes a fleet of aircraft with a limited number of types of Airbus.

The planning period is usually half a month or one complete month, within which the number of crew members with one crew position (e.g., captain or first officer) ranges from around 50 to nearly 200 for one fleet. The number of flights involved may grow to around 2000 for only half a month, depending on the size of the fleet.

Chapter 3

Literature Review

Optimization problems in large public transportation networks, such as airlines (both cargo and passenger airlines), railways and bus companies, are one of the major fields in operations research since about forty years. A high number of research articles and reports have been published over years to address the planning problems arising in these transportation networks. Notably, numerous planning and scheduling problems arising in scheduled passenger airline industry have drawn many researchers' attention for decades. As stated in Barnhart et al. (2003), the airline industry is the only sector, with possible exception of military operations, with which operations research has been linked so closely, because airlines provide a natural context for the application of OR techniques and models.

In fact, operations research has been one of the principal contributors to the enormous growth that the air transport sector has experienced during the past 50 years. Numerous articles (see Barnhart and Cohn, 2004; Etschmaier and Mathaisel, 1985; Gopalan and Talluri, 1998; Rushmeier et al., 1995) and recent books (Barnhart et al., 1999a; Yu, 1998) address many planning problems and related solution methods in airline industry. The problems, such as fleet assignment, aircraft routing, gate assignment, crew scheduling, etc, have been systematically studied in theory and practice.

In this chapter, an overview of solving the airline crew recovery problem is given from both theoretical and practical perspectives. In Section 3.1 we discuss literature on the airline crew scheduling problem, including the airline CPP and

CAP. In Section 3.2 recent research work on the airline crew recovery problem is presented in great detail. Finally, Section 3.3 contains a short summary of the literature review, and presents the general objectives of this research at the end of the chapter.

3.1 Review of the Airline Crew Scheduling

Crew scheduling issues involved in various transportation problems have attracted great attention of many researchers all around the world. Voluminous literature and reports appeared in the areas of bus drivers scheduling (e.g., Dias et al., 2001), railway crew scheduling (e.g., Caprara et al., 1998), and airline flight crew scheduling (e.g., Suhl, 1995). Crew scheduling problems in different areas have a certain similarity, whereas, each problem has its own distinctive characteristic leading to different dedicated solution methods.

Beasley and Cao (1998) discuss a generic crew scheduling problem that is clearly defined, and can be possibly applied in other particular industries. They consider the crew scheduling problem as a problem of assigning K crews to tasks with fixed start and finish times such that each crew does not exceed a limit on the total time it can spend working. Therefore, such a generic problem is formulated in a way that it tries to find K time limit constrained vertex disjoint paths which visit all vertices on a network. A lower bound is found via dynamic programming, and it is improved through a Lagrangean based penalty procedure and subgradient optimization. In their article, a number of randomly generated problem instances involving between 50 and 500 tasks are tested.

In practice, the problem becomes more involved and complex. It is one of the most difficult combinatorial problems which have been studied in scientific literature (Freling et al., 2001; Yunes et al., 2000). Early studies from 1960s, e.g., a survey from Arabeyre et al. (1969) already present approaches based on mathematical programming, including 0-1 integer programming, network flow approach, etc. However, at that time the solution methods, software and hardware were not able yet to cope with problems of practical dimension. The first commercial computer-based crew scheduling system was TRIP (Trip Reevaluation and Improvement Program) developed by IBM as reported by Rubin (1973).

Generally, the task of crew scheduling is to assign all flights of a given timetable together with further activities to a limited number of crew members stationed at one or several home bases. Besides the consideration of all given activities, operational cost has to be minimized, and workload should be evenly distributed among home bases and crew members.

An airline crew typically receives a monthly or semi-monthly schedule which has to fulfill numerous work rules and regulations. There is a bundle of rigid rules imposed by civil aviation authorities, union contracts, and company policies (see Barnhart et al., 1999b; Kohl and Karisch, 2002; Mellouli, 2003; Suhl, 1995, for example). Less rigid rules considering crew satisfaction and personal preferences can be applied as well. For these reasons, the problem becomes very difficult to solve, and more complex when problem size increases.

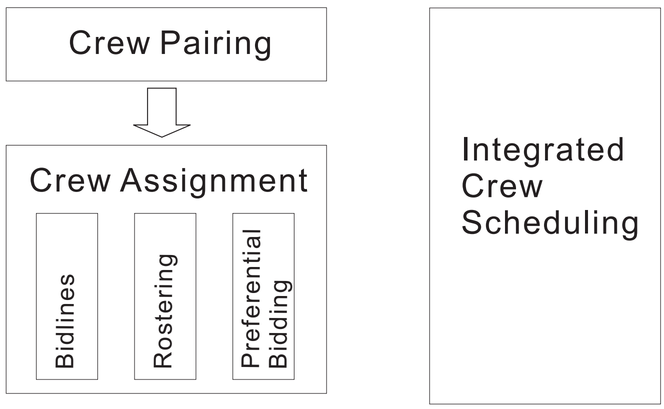


Figure 3.1: Airline crew scheduling approaches

Due to its complexity the CSP is typically divided into two sequential sub-problems (see Barnhart et al., 1999b), as depicted in Fig. 3.1:

- The airline crew pairing problem: Building a collection of crew pairings for all crews, such that each flight is covered by such set of pairings in a way that the underlying workforce demand is satisfied. The resulting set of pairing is optimized in terms of the achieving of minimum cost.
- The airline crew assignment problem: Constructing crew work schedules by chaining previously generated pairings into legal rosters (lines of work)

for a given planning period, and assign those to individual crew members considering their pre-scheduled activities and minimizing operational costs.

Previous research on the two sub-problems will be described in the following sections 3.1.1 and 3.1.2.

3.1.1 The Airline Crew Pairing Problem

Basically, the airline crew pairing problem (CPP) is the task of generating a set of pairings, which fulfills all the rules and regulations. The pairing set should include all or necessary pairings that may potentially be included in the final solution. And then a complete coverage of all flight legs examined must be projected after the selection of pairings. Most importantly, the cost of the total pairings should be minimized.

3.1.1.1 Problem Formulations

As one may observe in a number of papers, the airline CPP is usually formulated as a set partitioning problem (SPP) or set covering problem (SCP) (see Klabjan et al., 2001; Mingozzi et al., 1999; Wedelin, 1995, for examples). The problem, therefore, is to find a subset of pairings with minimal cost, in which every single flight leg is covered by exactly one chosen pairing (for SPP formulation, and being included by more than one pairing indicates the application of SCP).

One commonly studied SPP model is proposed by Barnhart et al. (1999a). It is expressed as follows

$$\min \sum_{p \in P} c_p y_p \tag{3.1}$$

$$s.t. \sum_{p\, i \in P} y_p = 1 \tag{3.2}$$

$$y_p \in \{0, 1\} \quad p \in P$$

where P (with index p) is the set of all feasible pairings constructed based on the set of flight legs F. The decision variable y_p is equal to 1 if the pairing p is included in the solution, and 0 otherwise. As shown in 3.1, the objective is basically the minimization of the cost of the set of selected pairings, while Equation 3.2 guarantees that every flight leg is covered exactly once.

In the case that SCP formulation is applied, the above equation 3.2, therefore, is changed to

$$\sum_{p\, i \in P} y_p \geq 1 \tag{3.3}$$

This restriction allows the possibility that more than one finally selected pairing include one single flight leg. To this particular problem, this modification can be understood as that crews can be deadheaded by means of planes. Deadhead is allowed in some airlines, but the number of deadheads is normally considered to be minimized.

For the case that multiple home bases exist, one additional constraint is usually added (see constraint 3.4).

$$l_{HB} \leq \sum_{p \in P_{HB}} y_p \leq u_{HB} \tag{3.4}$$

Let l_{HB} and u_{HB} be the lower and upper bound on the total number of crew members available at home base HB, respectively. P_{HB} is the set of pairings that have their first flight starting and last flight ending at home base HB. Notably, u_{HB} is usually much larger than the actual number of crew members stationed at home base HB due to the fact that one crew member may serve several pairings within the planning period examined.

Pairings are generated for crew members anonymously (see Barnhart et al., 1999a). A pairing is considered as legal as long as it fulfills some of the rules introduced in Section 2.2.3. Because of the anonymous generation of assignments, rules that explicitly consider individual crew members will not become applicable during the phase of the pairing generation. Follows are a list of rules that are usually examined for generating legal pairings:

- maximum daily/weekly flight hours
- maximum flight duty hours

- minimum off-duty interval
- maximum number of daily landings
- rest period between flights

When a pairing is created, these rules must be applied. By doing this, the legal pairings are selected and collected. Therefore, the task turns out to be how to generate such a set of legal pairings. Notably, the above SPP/SCP formulation requires the complete set of all possible pairings that has to be explicitly enumerated (Vance et al., 1997). Enumerating pairing can be an extremely complex task because of the large number of potential pairings and a number of rules that have to checked for each possible pairing. For example, a domestic problem in the U.S., on a hub-and-spoke network with several hundred flights, typically has billions of pairings (Barnhart et al., 1999a), let alone instances with around two thousand flights.

3.1.1.2 Solution Approaches

Relatively small instances may be applicable to be solved directly by a commercial or standard IP (integer programming) or MIP (mixed integer programming) optimizer, such as ILOG CPLEX (ILOG, 2002), or MOPS (Suhl, 1994, 2000) etc. But for practical problems it still is a formidable challenge for airlines to do. Therefore, an airline often starts with building flight duties, when a flight instance for a week or a day is being solved. With the help of such a set with a limited number of legal flight duties, the construction of pairings may turn out to be less difficult. Nevertheless, dedicated methods that show the great efficiency are desirable due to the increasing growth of airlines' scale and arising practical issues.

Since a set partition or covering model is commonly applied for solving such problems, methods that are applicable for SPP/SCP may have potential for finding "good" or even optimal solutions for the airline CPP. Some of them can be seen as exact methods that provide an optimal solution at the end, while others normally solve the problem heuristically.

Since the best known exact algorithm for linear integer programs is the Branch-and-Bound method (Ernst et al., 2004), many approaches are carried out based on the method of Branch-and-Bound. Generally, it starts with a solution to the linear relaxation of the original integer program. For each integer variable of the linear program which is not integer in the optimal solution of the relaxed problem, two options (branches) of rounding that value up or down are created as a constraint to a further linear program. In addition, the resulting solution is evaluated (bounding, including lower bound and upper bound), by which the number of branches can be reduced. The procedure is repeated until an optimal integer solution is found. Various branching strategies are widely adopted within the process. For further details, we refer to Anbil et al. (1992).

Andersson et al. (1997) report that the gap between the optimal objective of the continuous relaxation and the optimal objective of the integer program is very small, based on computational experiments. And small instances often have integer solutions but may have a gap of up to a few percent. However, larger problems rarely have integer solutions to the continuous relaxation, but the gap is always extremely small. For example, it is also mentioned in the article that a study conducted based on over one hundred large instances from several European airlines, the gap is almost always less than 0.5% and typically 0.1% for the largest problems.

When the size of the problem increases, explicit enumeration of all possible crew pairings becomes more and more intractable. Enumeration of crew pairings becomes the most time consuming task, which makes those algorithms impractical because of the explicit enumeration. Recent approaches follow the idea of column generation (Anbil et al., 1998; Crainic and Rousseau, 1987; Desrosiers and Lübbecke, 2003; Lavoie et al., 1988). Its basic idea is to implicitly consider all possible pairings (columns) by pricing out "good" ones, namely explicitly generating only a small subset of them. A small initial set of pairings is generated (e.g., in a heuristic way), which is treated a basis. By doing this, it is possible to consider a *restricted master problem*, the LP relaxation of the corresponding integer program, instead of the problem with all columns. After the restricted master problem has been solved, the resulting optimal dual vector π is used to find new columns that have negative reduced cost. The procedure is repeated until no more columns with negative reduced cost can be found. It is proven that the optimal solution of the restricted master problem is the optimal solution of the linear relaxation of the original integer problem with all possible variables. The optimal integer solution can be found by applying e.g., a branch-and-bound method (Anbil et al., 1994).

The subproblem turns out to be the generation of columns, which is usually formulated as a *resource constrained shortest path problem*. Therefore, a network is constructed either by building arcs from flight legs (Desrochers and Soumis, 1989; Desrosiers et al., 1991, 1995; Minoux, 1984) or duty periods (Anbil et al., 1994; Barnhart et al., 1994; Lavoie et al., 1988; Vance et al., 1997). The significant difference between the two network representations can be seen as large differences in network sizes and in the number of labels needed in the multilabel shortest path algorithm (Barnhart and Cohn, 2004).

Hoffman and Padberg (1993) propose a branch-and-cut method to solve the problem optimally as a set partitioning problem with side constraints. They generate cutting planes based on the underlying structure of the polytope defined by the convex hull of the feasible integer point and incorporate these cuts into a tree-search algorithm that uses automatic reformulation procedure, heuristics and linear programming to assist in the solution. Various experiments are conducted based on a large number of instances from several North American airlines. Great cost saving is also reported in the article.

Lately, a number of approaches fall into the class of *branch-and-price*. In short, branch-and-price dynamically applies a column generation procedure throughout the branch-and-bound tree. The major difference to traditional branch-and-bound is that column generation is applied to solve LP relaxation at each node of the branch-and-bound tree to create bounds. Recent development can be found in Barnhart et al. (1998), Desaulniers et al. (1998) and Freling et al. (2001)

Alternatively, network flow models can be applied in a number of approaches. Yan and Tu (2002) introduce a network model to improve the efficiency and effectiveness of solving China Airlines crew scheduling problems using real-life constraints. The problem, with relatively simple work rules, is formulated as a pure network flow problem. The network simplex method is used to solve the defined problem. Computational results are reported based on a number of instances representing particular routes.

A so-called *state-expanded aggregated time-space network flow* approach is proposed by Guo et al. (2003) and Mellouli (2001, 2003). The basic idea of applying states used for the vehicle maintenance routing problem is adopted to solve the airline pairing problem, precisely the airline crew pairing chain problem (as defined in the articles above).They observe that aircraft or trains are routed in order to regularly pass through a maintenance base, e.g., every three to four operation days for inspection. Likewise, crews are scheduled so as to "pass through" their home bases on a regular basis, e.g., weekly rest after five working days. This analogy is utilized to solve the crew scheduling problem. A mixed-integer flow model based on a state-expanded aggregated time-space network is developed. The mathematical model, formerly used to solve large-scale maintenance routing problems for German Rail's Intercity trains, is then extended to the airline crew pairing problem where "maintenance states" are replaced by "crew states". The advantages of such resulting network flow approach include the consideration of crews' time-dependent availability and the network with multiple crew home bases.

Besides the wide application of exact optimization methods introduced above, various heuristics, such as constructive, local-search based, and evolutionary heuristics etc, have also been used for the airline crew scheduling problem. Heuristics are widely applied in most of airlines today due to the nature of the simplicity and the performance. For example, early research done by Baker et al. (1979) describes a dedicated heuristic set covering algorithm, which gradually improves the solution. Several solution improvement procedures are also presented, e.g., an 2-opt algorithm.

Meta-heuristics, e.g., tabu search (TS), simulated annealing (SA), genetic algorithm (GA) etc, are also studied by many researchers recently. Emden-Weinert and Proksch (1999) report their experience of applying a simulated annealing algorithm to solve the airline crew scheduling problem, more precisely the airline crew pairing problem. In the article they propose a run-cutting formulation which models the cutting of segments from aircraft rotations and pasting them together to form pairings. The linkages between pairings and crew home bases are created, representing the minimization of global proceeding and hotel costs in view of the distribution of crew members over a couple of home bases. Computational results are reported for some real-world short- to medium-haul test problems with up to 4600 flights per month.

Cavique et al. (1999) introduce a tabu search based algorithm for solving the crew scheduling problem, in which an effective ejection chain method and the oscillation strategy are applied.

Timucin Ozdemir and Mohan (1999) present a graph based genetic algorithm that adopts a new graph based representation which demonstrates efficient memory usage. Various operators are applied in their approach, e.g., recombination operators (*set based operator*, *time based operator* and *distance preserving operator*) and the mutation operator.

Due to the large number of flight legs, the computational time of solving one specific crew pairing problem is comparably long even though efficient algorithms are already applied. Therefore, many researchers focus on parallelized algorithms to take over the task that is computation intensive. Details of various parallel algorithms can be found in e.g., Alefragis et al. (2000), Klabjan and Schwan (2000) and Sanders et al. (1999).

It is remarkable if research has practical applications. Scientific research results may help in solving problems that are too complex and difficult to resolve traditionally. Various applications in airline industry have been reported in scientific literature, particularly the airline crew scheduling problem. In Andersson et al. (1997); Hjorring and Hansen (1999); Hjorring et al. (2000); Karisch (2003), the authors present the pairing construction system, commercialized by Carmen Systems AB, which is in operation at most major European airlines. Firstly, a so-called *pairing generator* is used to create a set of pairings. It is basically a depth first search procedure in a search tree determined by the connection matrix representing all legal connections between flight legs. Secondly, a *cover optimizer* is applied to handle huge amount of pairings generated. The optimizer is an approximation algorithm for solving such large 0-1 integer programming problem, which is fine tuned to meet practical needs. This makes it possible to solve large practical instances. More applications of the airline crew pairing problem can be found in Anbil et al. (1991).

3.1.2 The Airline Crew Assignment Problem

The airline crew assignment problem (CAP) focuses on assigning pairings to *lines of work* and takes into account the need to provide sufficient rest periods between flights and to satisfy regulative requirements and collective bargaining agreements (Barnhart et al., 1994). It is the procedure of creating lines of work within a period of half or one month, linking pairings with pre-scheduled activities, such as training, vacations, requested off-duty periods, rest period etc.

In other words, it is the process of the "personalization" of schedules for each individual crew member by taking into account his/her availability and scheduled activities. that the goal is that the resulting schedules require minimal cost, and all flights and other activities are correctly served. The result of the assignment expresses the complete work that a crew member undertakes for the next half of or one month.

The assignment is the subsequent step that follows the airline CPP previously introduced in Section 3.1.1. As the pairings generated within the CPP are used for the CAP, the two concepts are interdependent. As described previously, due to the complexity the total airline crew scheduling problem is divided into two steps. As a consequence, the second step, the airline CAP, works at the level of pairings rather than flight legs, which reduces difficulty of the problem significantly.

3.1.2.1 Characteristics of the Crew Assignment Problem

Despite the fact that the task of CAP stated above stays the same for most airlines all around the world, there are several ways to cope with it. Various factors, such as the payment system, work rules, quality-of-life etc, lead the difference of assignment approaches (see Fig. 3.1). In North America, flight crews are able to bid their flying schedule, called *bidline*, for next month based on their seniority. A bidline generation approach constructs anonymous cost-minimizing bidlines and then lets individual crew members express their preferences through a bidding process (Campbell et al., 1997; Christou et al., 1999).

However, the seniority principle does not apply in European airlines (see Kohl and Karisch, 2004). The *personalized rostering* approach, or assignment approach, usually constructs personalized lines of work for each individual crew member by taking into account his or her working contract, pre-scheduled activities and requests. Another major concern is the even distribution of the workload among all crew members examined, namely the fairness for all crew members takes priority over others.

Apart from rules that are mainly considered in the previous pairing generation step (see Section 3.1.1.1), the following rules are the most important ones presented in most existing airline crew assignment approaches.

- maximum monthly/yearly flight hours
- maximum time away from crew's home base
- minimum monthly flight hours
- minimum weekly/yearly rest

Recently, preferential bidding becomes more and more the approach selected. Generally speaking, it represents a compromise between the bidline and rostering approaches in that it generates personalized schedules simultaneously taking into account a set of bids that have been weighted to reflect the employees' preferences. Gamache et al. (1998) present a preferential bidding system that is used at Air Canada since May 1995. They generate a set of weighted bids that reflect individual preferences of each crew member. The assignment is carried out under strict seniority restrictions: The construction of a maximum-score schedule for a particular crew member must never be done at the expense of a more senior employee. For each employee, from the most senior to the most junior, an integer model is solved to determine the crew member's maximum-score schedule while taking into account all the remaining crews until the entire problem is solved. The solution of such an integer model is generated by a column generation procedure embedded in a branch-and-bound tree. Further similar approaches can be found in Achour et al. (2003), Campbell et al. (1997) and Jarrah and Diamond (1997).

The reason for these different assignment "philosophies" lies in the differing nature of working contracts in various parts of the world (Doerner et al., 2003). Taking airlines in North America as an example, work schedules involving high workload result in high pay and vice versa. In short, crews are highly paid if they fly more. By contrast, flight crews in Europe work under their work contracts that guarantee fixed payment for a certain minimum amount of flying hours (so-called *block hours*), no matter whether a crew member actually performs given flight hours or not. This makes airlines in Europe concentrate on producing crew schedules whose workload should be higher than the guaranteed lower bound written in crews' contracts. At the same time, the operational cost is minimized.

In this thesis, we mainly study the assignment task that is normally examined in European airlines: the personalized rostering problem. Therefore, the discussion of the airline crew rostering problem will dominate the rest of this section. In Section 3.1.2.2, the commonly applied mathematical model is introduced. Various solution approaches are discussed in Section 3.1.2.3.

3.1.2.2 Problem Formulations

In this section, we give a basic mathematical model proposed by in Barnhart et al. (1999a) and Gamache and Soumis (1998). Let set P be the resulting set of pairings from the process of pairing generation. We are given a set of crew members W, and a set of activities A that represent all pre-scheduled activities. The set R denotes all feasible rosters that can be assigned to each specific crew member. The task, therefore, is to find the subset $\bar{R}$ which represents the partition covering all $p \in P$ and $a \in A$. Most importantly the total cost of assignments is minimized. Therefore, the model is built as

$$min \sum_{w \in W} \sum_{r \in R^w} c_r^w x_r^w \tag{3.5}$$

$$s.t. \sum_{w \in W} \sum_{r \in R^w} \gamma_p^r x_r^w \geq n_p \qquad \forall p \in P \tag{3.6}$$

$$\sum_{R \in R^w} x_r^w = 1 \qquad \forall w \in W \tag{3.7}$$

$$x_r^w \in \{0, 1\} \qquad \forall r \in R^w, \forall w \in W$$

where R^w denotes the set of feasible roster of the crew member $w \in W$, and n_p represents the minimum number of crew members required by the pairing p.

Binary value γ_p^r is 1 if pairing p belongs to roster r and 0 otherwise. The decision variable x_r^w equals 1 if roster r is assigned to crew member w and 0 otherwise. c_r^w expresses the cost of assigning a roster r to crew member w (the calculation of the cost may differ from airline to airline). As one may observe, the objective function 3.5 is the minimization of the sum of all costs for every roster examined. The constraint 3.6 guarantees that each pairing p is served by the required number n_p of crew members, while 3.7 makes sure that each crew member w receives exactly one roster for the given planning period.

Generally, the cost c_r^w may be the combination of the real operational cost and other artificial cost, such as the transformed monetary cost of crews' quality-of-life. Comparing with the airline CPP, the cost in CAP, therefore, may be difficult to calculate due to aspects that are not well-defined. In addition, the real cost discussed here is not the same as that observed in North America because of the application of the rostering principle in European airlines. In Europe, the real cost normally refers to the money paid for transits and hotel stays instead of crews' salary.

The airline CAP is normally decomposed by crew functions. This is particularly the case when a rostering problem for cockpit is examined. Accordingly, it can be divided into two subproblems, for captain and first officer respectively (e.g., for short-haul cockpit crew scheduling problem). In such case, the formula 3.6 may be rewritten as

$$\sum_{w \in W} \sum_{r \in R^w} \gamma_p^r x_r^w = 1 \qquad \forall p \in P \tag{3.8}$$

The advantage of this decomposition is obvious because the complexity of the problem is reduced significantly. Therefore, the possibility of finding better or even optimal solution is significantly raised. On the other hand, this decomposition may have certain drawbacks in the case that aspects, such as *downgrading* and *team* principle, are considered. The purpose of downgrading is to fill in positions required for lower ranked crew by higher ranked crew, further details we refer to Dawid et al. (2001) and König and Strauss (2000). For increasing work efficiency and safety, team building is concerned by airlines. A thorough discussion of team aspects can be found in Thiel (2004).

3.1.2.3 Solution Approaches

Essentially, the airline CPP and CAP have a great similarity. The basic mathematical models, namely SPP or SCP, are analogous to each other except that CPP is dealing with pairings but CAP takes care of rosters. Another evident difference lies in the phase where pairings or rosters are generated, because they have to fulfill a certain number of work rules and restrictions. Consequently, to some extent the methods applied for solving the airline CPP can also be possibly used to tackle the airline CAP, such as constructive heuristics, column generation, branch-and-price and so on (see Kohl and Karisch, 2004, as an overview).

Therefore, in this section we do not introduce the basic ideas of some particular algorithms that are already mentioned in Section 3.1.1.2. Instead, we summarize it in a way that algorithms appearing in literature are singled out and distilled into following groups (the revision of the work by Gamache et al., 1999):

1. Constructive heuristics that construct rosters gradually.

   (a) Rosters are built by assigning high-priority activities to high priority employees (Glanert, 1984; Marchettini, 1980).

   (b) Rosters are built by assigning pairings day by day. For each day of the roster period pairings are assigned to individuals that are chosen from a pool of available crew members (Buhr, 1978; Nicoletti, 1975; Sarra, 1988; Tingley, 1979).

   (c) Monthly rosters are constructed for individual crew members sequentially, starting with those with higher seniority (Byrne, 1988; Moore et al., 1978).

   (d) The combination of two methods above, where rosters are first constructed sequentially for each crew member, and then reoptimized day by day (Giafferri et al., 1982).

2. The simulated annealing algorithm is also widely studied for crew scheduling. For example, in Lučić and Teodorović (1999) a simulated annealing algorithm is developed to improve the solution that is created by the *pilot-by-pilot* heuristic initially.

3. The approach from El Moudani et al. (2001) uses genetic algorithms to generate new solution sets with reduced operational cost over a sequence of generations. A new mathematical formulation which takes into account the satisfaction of the crew members is proposed. A genetic algorithm based heuristic approach is adopted to produce reduced cost solutions associated to acceptable satisfaction levels for the crew staff. The application of the proposed approach to a medium-sized airline is evaluated. Timucin Ozdemir and Mohan (1999) also propose a genetic algorithm for solving the airline crew scheduling problem, in which a graph based representation is adopted. Marchiori and Steenbeek (2000) develop an evolutionary algorithm for large scale set covering problems, and the application to the airline crew scheduling is described. Other similar approaches can be found in Levine (1996), Kerati et al. (2002).

4. A generalized set partitioning model is used to solve the rostering problem. A heuristic first produces a prior set of feasible rosters for each crew member and then constructs a constraint matrix that helps the search for an integer solution. Then the utilization of specialized integer programming solves the rostering problem. For example, Ryan (1992), Ryan and Falkner (1988) and Butchers et al. (2001) present details on linear relaxation and branch-and-bound technique. A further extension to handle the downgrading issue is added by Dawid et al. (2001).

5. A 0-1 multicommodity flow model is built by Cappanera and Gallo (2001, 2004), in which each crew member represents a commodity in the network. Several small instances from a medium-sized Italian carrier are solved with the CPLEX MIP solver.

6. Column generation is applied within the branch-and-bound scheme. The subproblem, generation of columns, is solved as a constrained shortest path problem (Fahle et al., 2002; Gamache and Soumis, 1998; Gamache et al., 1999; Junker et al., 1999). In Yunes et al. (1999), Yunes et al. (2000) and Yunes et al. (2001), a hybrid column generation algorithm, combining constraint logic programming (CLP) and integer programming techniques,

is developed for solving several real-life airline crew scheduling problem instances. The performance of their algorithms is evaluated in terms of faster problem solving and better solutions found. The branch-and-price algorithm is described in Freling et al. (2001).

3.1.3 Integrated Airline Crew Scheduling

One important reason for adopting the two-step sequential approach is that it is usually impossible to solve the joint airline crew pairing and crew assignment/rostering problem in one step because of the high combinatorial complexity of both problems. For large practical cases, it is not even possible to find an exact optimum for any one of the two steps with current state-of-the-art technologies. However, a fully integrated approach to the airline crew scheduling remains undoubtedly difficult. Therefore, it remains an important and challenging research task to find out ways of partial integration of the two steps, so that the drawbacks mentioned above can be removed at least to a certain extent. Especially nowadays, the increasing computational power has made it possible to solve seemingly impossible problems observed in the past. In summary, the integrated airline crew scheduling approach will continue to be the future research trend.

Guo et al. (2003) propose a partially integrated procedure to solve the airline crew scheduling problem. They develop a special network flow model, called state-expanded aggregated time-space network flow model that generates not only pairings, but most importantly *pairing chains* as sequence of pairings which covers the scheduled time period, incorporating weekly rests so that all valid rules and regulations are taken into account. By taking guaranteed pre-scheduled activities of individual crew members into account, the real number of available crew members on each day – called dynamic crew capacity – can be exactly considered already in the pairing generation phase, thus improving the total solution quality.

Klabjan et al. (2002) propose a partial integration of crew scheduling and aircraft routing in which they consider the feasibility of aircraft routing by adding plane count constraints to the crew problem. It is reported that resulting solutions to the crew scheduling problem have significantly lower costs than those obtained from the traditional model. Cohn and Barnhart (2003) present the so-called extended crew pairing model which integrates key decisions of aircraft routing with crew pairing.

Similar integration ideas appear in other transportation areas as well, e.g., for vehicle and crew scheduling problem in bus companies. Freling et al. (2000) propose new mathematical formulations for vehicle and crew scheduling problems in a completely integrated fashion. In their approach, Lagrangian relaxation is addressed, together with an implementation using column generation applied to a set partitioning type of model. Based on the computational results tested on real life data, they analyze the performance of algorithms proposed by comparing with traditional sequential approaches. The applicability of the proposed techniques to practical integrated problems is approved.

3.2 Review of the Airline Crew Recovery

After the schedules have been planned, the operations phase is about to start. Unfortunately, one never knows what might happen in future, simply because many unforeseeable events or situations may occur. The uncertainty about the operation environment ahead forces airlines to make concerted efforts to recover from any disruptions that have happened to them.

During the *day of operations*, unexpected events keep happening from time to time as discussed in Section 2.1. A rescheduling activity is typically carried out by airline operations controllers who are typically located in the airline operations control center (see Clarke et al., 2002; Yu et al., 2003).

The recovery problem, also called disruption management, is usually composed of three processes (see Section 2.2.1). When an irregular operation occurs, some aircraft may be rerouted within an aircraft recovery process. Besides rerouting aircraft, decisions on delaying and cancelling flights are also made in this stage. It is followed by the process of crew recovery, where new itineraries may be assigned to crews (but not necessarily changed if possible). In order to reschedule crew, coordinators may use operating, standby, and reserve crews to cover all open flights. At the end is the passenger reaccommodation process, where passengers are rerouted to alternative itineraries. Clearly the new schedule must conform to all regulatory and contractual rules. Contractual rules for operations are usually different from those in planning. Notably, crew management during irregular operations is usually the bottleneck of the whole system-recovering process due to complicated crew schedules and restrictive crew legalities as well as the size and scope of the hub-and-spoke networks adopted by major carriers (Wei et al., 1997).

3.2.1 Problem Formulations

The detailed description of the airline crew recovery problem is already given in Chapter 2. Here we discuss a basic mathematical model proposed by Wei et al. (1997), which basically can be viewed as an integer multi-commodity network flow problem (3.9–3.12).

$$min \sum_{kj_k} c_{kj_k} x_{kj_k} \tag{3.9}$$

$$s.t. \sum_{kj_k} a_{ikj_k} x_{kj_k} \geq 1 \quad i = 1 \text{ to } n \tag{3.10}$$

$$\sum_{j_k} x_{kj_k} = 1 \quad k = 1 \text{ to } m \tag{3.11}$$

$$x_{kj_k} = \{0, 1\} \tag{3.12}$$

Here j_k is the j_kth pairing for crew member $k \in K$, and i is the index of flight leg set. a_{ikj_k} equals to 1 if flight i is covered by pairing j_k of crew member k, 0 otherwise. c_{kj_k} denotes the cost of assigning pairing j_k to crew member k. The decision variable x_{kj_k} is 1 if pairing j_k of crew member k is part of the solution, 0 otherwise. In this model, each crew member, including standby/reserve crew, represents a commodity. The first set of constraints (3.10) shows the coverage constraints, requiring that each flight in the network must be covered. The second set of constraints (3.11) denotes flow conservation, which restricts that one crew be assigned to only one pairing. The objective function (3.9) represents the minimization of the cost of assigning all pairings.

As one may see from the model above, all pairings have to be pre-constructed before the problem solving actually starts. Therefore, it can also be understood as a crew pairing repair approach. One advantage of such an approach is that it does not require too long time to solve, which is particularly important for a recovery problem in practice.

3.2.2 Solution Methods

Today the work in solving the airline crew recovery is still at the beginning stage. As delegates from major airlines all around the world gathered at AGIFORS Crew Management Study Group 2003 Conference, it was widely acknowledged that almost all airlines rely primarily on manual methods to face such challenges. They described the challenge of tackling such problems as their day-to-day work without a dedicated system that may fundamentally assist them to fix the problem.

Some researchers have conducted a number of experiments to solve irregularities occurred during airlines' daily operations. Many observations about the problem have been shown in research papers, including methodologies and various practical considerations. In this section, a literature review on the airline crew recovery problem is given, in which the latest solution methods and practices are reviewed based on previous research.

Abdelghany et al. (2004) present a practical application of a decision support tool that automates crew recovery during irregular operations for large-scale commercial airlines. The system adopts a rolling approach in which a sequence of optimization assignment problems is solved such that it recovers flights in chronological order of their departure times. In each of them, the objective is to recover as many flights as possible while minimizing total system cost resulting from resource reassignments and flight delays. The advantage of their approach over the existing ones is that it recovers projected crew problems that arise due to current system disruptions ahead of their occurrence. In addition, it gives a wide flexibility to react to different operation scenarios. A test case is presented to illustrate the model capabilities to solve a real-life problem for one of the major commercial airlines in the U.S.

Lettovský et al. (2000) proposed a pairing generation method with special branching strategies for solving the crew recovery problem. They build a pairing based model similar to the model normally observed in crew pairing problems. In their model, each pairing is specific to a particular crew, thus not anonymous as is the case for the crew pairing problem. The objective of the model is to minimize the cost of adjusted pairings, reserve crews, and deadheaded crews, as well as the cost of cancelling flights. The cancellation cost is the cost of reassigning passengers to other flights as well as hotel and meal costs for affected passengers and some estimate of the loss of good will.

Nissen (2003) presents a duty-period-based network model for solving the airline crew rescheduling problem. A network is built with nodes representing airports, and arcs representing duty periods. All possible duty periods are generated prior to the creating of the network. The solution method is basically a branch-and-price scheme, in which a column generation procedure is embedded into a branch-and-bound framework. The sub-problem of the column generation is tackled by the resource-constrained shortest path algorithm. Experiments are conducted based on two instances: short-haul and medium-haul. Both instances include only one hub, with 8 and 35 routes respectively.

In Stojković et al. (1998) authors present a column generation approach which is a slight deviation from the one that is used to solve the crew pairing problems. Basically the algorithm is designed to generate personalized pairings, and the assignment of them is carried out simultaneously. In short, they solve such problems as an integer nonlinear multi-commodity network flow model with time windows and additional constraints. *Dantzig-Wolfe* decomposition combined with a branch-and-bound method is proposed. The corresponding sub-problem of column generation is a constrained shortest path problem where a duty-period-based network is constructed for each crew candidate with duty periods represented as nodes. Computational results are reported testing problems with up to 16 crew candidates and 210 tasks in total with two different (1 or 7 days) operational period. Depending on problem instances, the solution times ranged from a few seconds to 20 minutes.

Stojković and Soumis (2001) describe and solve the operational pilot scheduling problem for one day of operations. They attempt to simultaneously modify the existing flight departure schedules and planned individual work days (flight duties) while keeping planned aircraft itineraries unchanged. The problem is addressed as the coverage of all flights for one day of operations with available pilots while minimizing changes in both the flight schedule and the next day's duties planned. The problem is mathematically formulated as an integer non-linear multi-commodity network flow model with time windows and additional constraints. To solve the problem, a Dantzig-Wolfe decomposition combined with a branch-and-bound method has been used. The master problem comprises the flight covering constraints and a new set of flight precedence constraints. Sub-problems consisting of time-constrained shortest-path problems with linear time costs are solved by a specialized dynamic-programming algorithm. Many tests are conducted based on several input data sets with up to 59 pilots and 190 flights in total, all of which could be solved in very short computational time.

A heuristic-based framework for handling disruptions is presented by Wei et al. (1997), in which a multi-commodity integer network flow model (see Section 3.2.1) and a heuristic search algorithm are developed to reschedule crew during irregular operations. The basic idea of their approach is to repair the broken pairings due to the modification of schedules that is mostly caused by the aircraft recovery. The primary goal of the approach, therefore, is to return the entire operation to its original schedule as soon as possible in a cost-effective way. Based on their quantitative analysis, a depth-first branch-and-bound search algorithm is in essence devised and implemented. A generic state representation of the problem is defined which characterizes each node of the search tree. At each node, the problem is represented by a set of uncovered flights and a list of pairings that are modified so far in the search process. The legality of each pairing is checked through a legality checking module that is invoked after the pairing generation and modification. Through the search process, a list of solutions is saved and is updated whenever a new and better solution is found. Further development of such approach can be seen in Song et al. (1998).

Yan and Lin (1997) describe the rescheduling problem caused by the closure of airports, not particularly for the rescheduling of crew. The problem is formulated as a pure network flow problem with side constraints, and solved by using the network simplex method and a Lagrangian relaxation based algorithm. A case study is given in the article which is based on real life data from China Airline's international operations. However, problem instances examined in the article are relatively small.

Yu et al. (2003) report several stories of successful recovery of some disruption scenarios, such as snowstorms and the September 11th terrorist attack. In the article they describe an award-winning real-life application employed by Continental Airlines in the U.S., in which the problem is treated as a set covering problem and a so-called *generate-and-test* heuristic is applied to generate rosters. An initial problem is first converted into a generic one by collecting the uncovered flights and repairing the broken pairings. A generic network is constructed in the same way as proposed by Wei et al. (1997), which may encourage the algorithm (*negative-cost shortest-path algorithm*) to assign crews to their originally assigned flights (as arcs). Pairings are regenerated if an uncovered flight is assigned to a given crew member, which is a similar process to the converting procedure, and uses the same network and algorithm. Once pairings have been generated, their legality is checked by an isolated component, the legality-checking module.

Issues regarding airline irregular operations are also discussed in Clarke (1995), Clarke et al. (1996), Clarke (1997), Clarke (1998), Clarke et al. (2002), Irrgang (1995), Rosenberger et al. (2002) and Rosenberger et al. (2003). In addition, the aircraft recovery problem is presented in Bard et al. (2001), Løve et al. (2001) and Thengvall et al. (2003), and the description of flight rescheduling problem can be found in e.g., Stojković et al. (2002).

Undoubtedly, further development in the subject of airline crew recovery will continue to be carried out. Many research groups and commercial solution providers have been deeply involved, and start to make great effort to develop dedicated software. For a further overview and a survey, we refer to Barnhart et al. (1999a) and Filar et al. (2001) respectively.

3.3 Summary

The state-of-the-art research work in the area of airline crew scheduling and rescheduling is pictured in this chapter. It shows the substantial contribution made by many researchers since decades. However, there is still much room left for further developments, especially regarding the newly emerged topic — the airline crew recover problem. In addition, the existing work was mainly carried out in the circumstances that only fit to the operation environment in North America. Much research is still needed to be conducted under the setting of European airlines. Accordingly, our research is highly motivated, and attention is paid to this particular subject.

Chapter 4

Mathematical Programming and Optimal Recovery Solution

As already discussed in Chapter 3, many optimization problems arising in large public transportation networks can be considered as planning problems in general, therefore methods for planning problems can be adopted for the problem examined in this thesis. For a long time various types of solution methods have been discussed in the areas of operations research and heuristics. Planning problems from different areas, such as airlines, railways and bus companies, have a certain degree of similarity in terms of their problem structure and corresponding solving methods.

In this chapter, mathematical formulations of the airline crew recovery problem are presented in Section 4.1 and 4.2. Later in Section 4.3, the exact optimization methods are described in detail, which attempts to solve such problems to optimality. Computational results tested on real-life instances from a European tourist airline are reported in Section 4.4. Finally, a brief summary regarding mathematical solutions of the problem is given in Section 4.5.

4.1 General Requirements

Prior to the mathematical formulation, some basic concepts and characteristics of the airline crew recovery problem are given in this section and followed by the mathematical model presented in Section 4.2.

4.1.1 Cost Minimization

Optimization of planning problems in the airline industry is often highly motivated by the potential cost savings, i.e., the cost saving factor usually comes as the first motivation for most relevant research in this area. Accordingly, the cost is normally considered as the most important factor when a planning problem is modeled mathematically. The airline crew schedule recovery is no exception and needs to be cost effective as well. It is not acceptable if a recovery solution in a disrupted situation costs too much additional money. For example, a recovered crew schedule may be technically feasible, with even distribution of workload among crew members examined, but requires much additional transferring between airports. It is also unacceptable if there are too many crew members who have to be notified because of changes, even though the recovered crew schedule does not need any additional operational cost.

Therefore, the minimization of additional operational cost $\bar{C}$ can be seen as

$$\{minimize(\bar{C}) \mid \bar{C} = C - C_{org}\} \tag{4.1}$$

where $\bar{c}$ is the result of subtracting the operational cost for originally planned crew schedule C_{org} from the updated operational cost C. This cost constitutes the major part of the objective. Given a certain problem instance and a planning period, the original operational cost of scheduling a set of crew members, C_{org}, is constant and can be computed directly based on the results produced by the planning system (airline crew scheduling system in the planning phase). Consequently, the minimizing of additional operational cost can be transformed into the minimization of updated operational cost C as shown below

$$minimize(C - C_{org.}) \Rightarrow minimize(C) \tag{4.2}$$

4.1.1.1 Operational Cost

Generally, the significant difference between airline crew scheduling (and rescheduling) approaches in Europe and that in the U.S. is the payment system. In Europe, flight crews are guaranteed a certain salary which represents a minimum number of block hours. However, in North American airlines crews are compensated based on the number of working hours, which might significantly deviate between crew members and time periods. That is to say, fixed salaries for crews are predominant in most European airlines, which differs from the payment structure appearing in most published literature that address the same or similar crew recovery problems in North America. This expresses the fact that operational cost dominates the cost structure to a certain degree. It basically includes the cost of transferring crew members and accommodating them when staying outside their home base overnight. The operational cost C_{opl} can, therefore, be calculated as

$$C_{opl} = \sum_{i \in R} \sum_{\forall\ trs} c^i_{trs} + \sum_{i \in R} \sum_{\forall\ htl} c^i_{htl} \tag{4.3}$$

where c^i_{trs} denotes the cost of a transit which takes place in roster $i \in R$. c^i_{htl} expresses the cost of a overnight stay at a hotel, which is included in roster $i \in R$. The operational cost in total is the sum of costs occurred for every transit and hotel stay of all rosters examined. The set of all rosters is collected over all home bases examined. The values of c^i_{trs} differ from one another, as they denote the transit cost from one city to another. It depends on the distance between the two cities and the way how the crew member is transfer — by train, taxi or airplane. Likewise, the value of c^i_{htl} may also vary, depending on the rate that the chosen hotel charges. Usually, only a certain number of airports are considered as hotel bases (e.g., over 20 hotel bases are available in the setting of the airline examined in our approach).

There are limits to introduce transits between airports instead of considering all possible transits between two airports. Transferring a crew member from one airport, e.g., one home base inside Germany, to another one that is located somewhere in Spain is in no way reasonable if there are other cheaper options. Normally, the creation of transits depends on the physical distance between cities, and also other public transport systems that can be taken. For example, transit by train might be much cheaper than taking a taxi, but only if there is no time pressure for the crew member. A comparably long time is necessary for crews if they take a train, because they need some time to get to the airport where they will serve their next flight.

Taking a taxi, however, also requires certain concerns, as it is very expensive. Some European airlines have set up particular rules regarding this issue. For instance, one rule may be expressed as: From airport A to B, it is allowed to take taxi only if the number of crew members who need to be transferred from A to B exceeds 8. Therefore, only one or two taxis are needed instead of calling one for each crew member. It is rather useful for flight attendants due to the fact that they usually work as groups. However, it is not very likely that enough pilots may share one taxi from one airport to another. Furthermore, this makes the problem more complicated because it is usually decomposed into the level of fleets and crew functions. Due to the focus of this work on rescheduling pilots (cockpit crews), we do not consider to form a certain number of crews before creating a transit. But, as desired by some airlines we limit the possibility of transits to a subset of all possible pairs of cities, say home bases together with those cities that are very close to them.

4.1.1.2 Cost For Using Standby/Reserve Crew

Apart from transfer and hotel cost mentioned above, the additional cost incurred by calling reserve crews is also another big concern expressed by airlines. It is another major difference between the process of crew rescheduling and the scheduling process in the planning phase. A cost is then added if a standby or reserve crew member is used to serve one or more flights. In this work, we consider such cost as a constant penalty for each assignment to a standby or reserve crew member. The penalty for the use of standby/reserve crews is then set to a value that is nearly the same as the estimated cost.

4.1.1.3 Change Cost

Another most desired feature is to diminish the variation from the schedule originally planned. As a certain subsequent action must be taken directly after a change occurs, the decision of changing original assignment may possibly cause further troubles, such as missing crews due to unexpected changes to him/her. Likewise, a single change may also cause complaints and put considerable inconvenience to affected crew members.

Basically, the cost of variations (or changes) from the original schedule c_{chg} can be computed as

$$C_{chg} = \sum_{i \in R} \sum_{f \in F(i)} c_{chg}^{f} \tag{4.4}$$

which is the sum of change costs c_{chg}^{f} over all rosters in the final solution. It can be calculated in a way that a certain amount of penalty is set to each change occurred. Such cost may be expressed in monetary form, so that it can be modeled directly. In a flight level, a direct change is introduced whenever it is reassigned to a different crew member other than its original owner. However, changes also differ greatly to one another, because of their distinctive patterns in terms of locations in the schedule. In our approach we take three different cases into consideration:

- A constant penalty $P1$ is imposed when the change is covered by the disrupted period (see Section 2.2.5 and 4.1.2) and the crew member who is chosen to operate the flight in the end is not originally affected by disruptions. In other words, his or her original schedule has nothing to do with the newly updated flight schedule, i.e., there is no intersection with the set of updated flights. It is simply because there is no direct need for this specific crew member to change his or her original schedule and it is also very likely that the crew member does not like such changes (sometimes, no matter what changes). This may, in turn, be one source of complaint, although crews seldom reject schedule changes.

- No penalty is introduced if the crew member who takes the flight in the final solution is originally and directly affected by disruptions. If a crew member is originally affected by a disruption, it can be understood that his or her original assigned flight is updated and different from the old one in terms of the schedule. Consequently, it is in no way possible that the crew member can keep his or her original schedule rather than having a new schedule.

- Another relatively higher penalty is added if the change is not covered by the disrupted period but inside the recovery period. As one can see in (4.5), it is calculated based on how long the change is away from the disrupted period. d is the number of days away from the disrupted period. D is the total examined time period in days and $P2$ is a constant value that is estimated for the change. By using $s1$, we are able to penalize more if airlines prefer keeping changes inside the disrupted period. The exponent $s1$ is set to 1 in our approach.

$$c_{chg}^{f} = \begin{cases} 0 & \text{if the change is in the disrupted period, and} \\ & \text{the crew member is originally affected;} \\ P1 & \text{if the change is in the disrupted period, but} \\ & \text{the crew member is not originally affected;} \\ (\frac{d}{D})^{s1} \cdot P2 & \text{otherwise} \end{cases} \tag{4.5}$$

Another possible way to minimize changes is to decrease the number of notifications. Due to the change of schedule, airlines have to inform their crew members involved about the changes immediately after the new schedule has been created. A solution that requires too many notifications is certainly not desirable. We can distinguish these different cases as:

- A notification is not desirable if it is about a change that will occur within the disrupted period but the crew member is not originally affected by disruptions.
- It is comparably easy to accept changes if the crew member is originally and directly affected by disruptions. It will also not cause further troubles in terms of unable to contact him or her, because he or she normally expects changes in such a particular situation.
- The number of notifications that inform crew members some changes on the next day or a few days later should also be minimized.

Basically, the goal of minimizing changes can be achieved in both ways. We choose the first approach due to the fact that it is more flexible to control changes in flight level.

4.1.2 Recovery Period

Due to the nature of the airline crew recovery problem, we usually consider only a shorter recovery period than is used within the planning process. The length of the recovery period chosen may have a great impact on the general performance of the recovery. First of all, it may influence the quality of the final recovery solution. Secondly, it may have an impact on the time that the problem solution procedure takes.

As described in 2.2.5, two types of periods are basically involved in this problem: disrupted period DP and recovery period RP. Here we make the assumption that some flights in the problem instance examined have been updated. Namely, their departure or arrival times are changed, or flights are newly added or cancelled due to current disruptions.

In order to determine the two periods, we first give following definitions:

$t_{dp.start}$	the departure time of the first flight that is updated due to the disruptions and has not been operated yet
$t_{dp.end}$	the arrival time of the last flight that is updated due to the disruptions and has not been operated yet
$t_{rp.start}$	the starting point of time where flights operated later than this point may be possibly rescheduled. Flights operated earlier than this point remain intact
$t_{rp.end}$	the ending point of time where flights operated earlier than this point, but later than $t_{rp.start}$, may be possibly rescheduled. Flights operated later than this point remain intact

Based on the description above, we get the sequence of the four points of time:

$$t_{rp.start} \leq t_{dp.start} < t_{dp.end} \leq t_{rp.end}$$

Therefore, the disrupted period and the recovery period can be calculated as shown in Equation 4.6 and 4.7 respectively.

$$DP = t_{dp.end} - t_{dp.start} \tag{4.6}$$

$$RP = t_{rp.end} - t_{rp.start} \tag{4.7}$$

Notably, the significant difference between the two periods is the fact that the disrupted period is fixed, while the recovery period is open rather than decisive. The recovery period is determined in respect of the specific need that reflects the given disrupted situation. Airlines may need a longer recovery period in order to deal with serious disruptions. In contrast, it might be shorter for a larger fleet with many crew members and flights involved, because the larger size of the problem instance normally requires a longer computational time to solve the problem.

It is an interesting question whether the starting time of the recovery period $t_{rp.start}$ should be exactly the same as the starting time of the disrupted period $t_{dp.start}$. The answer that we propose is that the starting time relies on the situation examined. One reason that $t_{rp.start}$ is earlier than $t_{dp.start}$ is that the problem can be relaxed, since it provides more possibilities of finding a better solution. Another reason to set an earlier starting time of recovery period is that disruptions are detected ahead of their actual occurrences and airlines can be able to handle the problem proactively.

4.1.3 Active and Frozen Flights

Once the recovery period RP has been determined, we may divide the flight legs into two groups: *active flights* and *frozen flights*. Flight legs that are scheduled to be operated earlier or later can be considered as "frozen" because of unnecessary modifications. In contrast, those flight legs that are operated within the recovery period are seen as "active" flights, which indicate the possible reassignment to other crew members. The reason is that the recovery period cannot be too long since the problem has to be resolved in a reasonable period of time. Therefore, a shorter one is set up as the "working" period for the crew recovery process. Within the period, we may assume there are sufficient flights and crew members to find a solution that withstands the given disruption.

As illustrated in Fig. 4.1, active and frozen flight legs can be determined. It is quite obvious that flight legs must belong to the group of active flights if both departure and arrival time of that flight are within the recovery period.

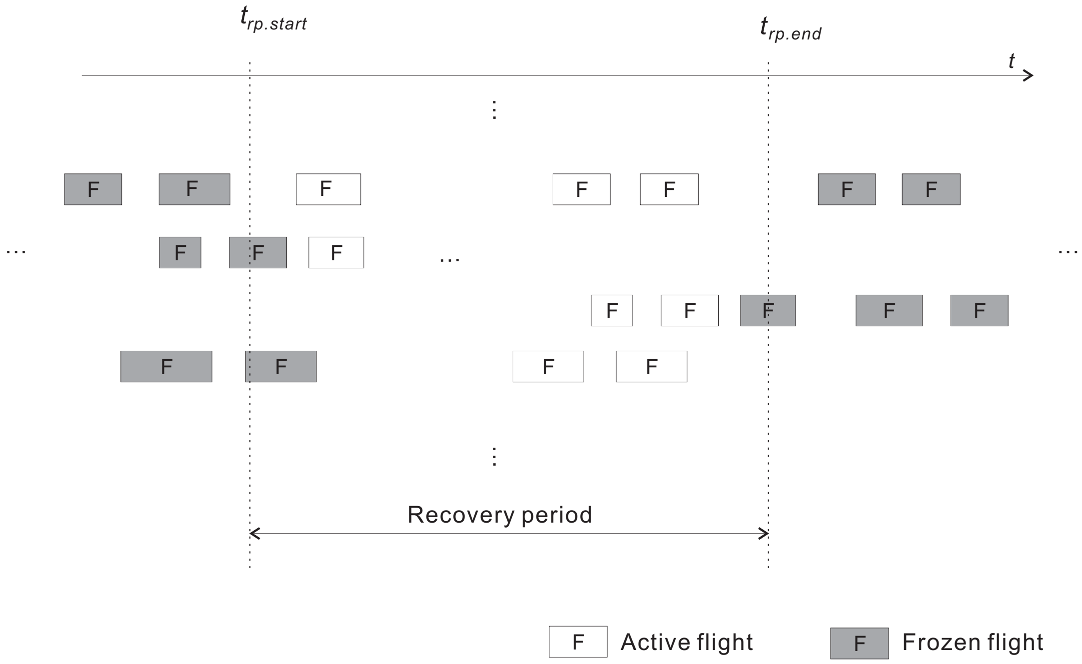


Figure 4.1: Active and frozen flight legs

Furthermore, a flight leg is also considered to be frozen if only one of the departure/arrival times is in the recovery period. For example, a flight that departs slightly earlier than the beginning of the recovery period but arrives later than it must be treated as frozen because there is no way to reassign such a flight to any other crew member. In other words, it is not necessary to consider the rescheduling of such a flight.

4.1.4 Decomposition

In the setting of the airline involved, crew members, particularly cockpit crew, are qualified to operate only a limited number of aircraft types. Airlines group their aircraft into fleets regarding an aircraft's generic specification. Therefore, in our approach we may decompose the problem and examine it fleet by fleet. This reduces the complexity of the problem dramatically. Moreover, crew positions (captain, first officer, second officer etc.) are usually not interchangeable, thus further decomposition can be made by separating an airline's crew positions.

Notably, more than one crew position is also considered in the case that a certain level of teamwork is required, e.g., the team may be built among captains and first officers. However, this is not in the scope of this approach, and the team issue is not considered in this work.

4.2 Set Partitioning Models for Airline Crew Recovery

After giving the general requirements of the problem, in this section the corresponding mathematical models are presented in detail. We discuss two models where the second model is a revised version of the first one, and show their different perspectives.

4.2.1 Basic Model

Similar to the airline CSP, the airline CRP can be mathematically formulated as a set partitioning type model, where a set of affected flights caused by disruptions needs to be assigned or reassigned exactly once. The disrupted flights grouped with previously planned flights are chained into a huge amount of rosters, which represent all possible individual schedules for crew members within a certain time period. Each crew member, therefore, will be finally assigned to at most one revised schedule for the examined period with respect to all the regulations and rules.

In our approach, we apply the concept of integration. Due to the shorter recovery period the problem is solved in an integrated way instead of addressing pairing generation prior to the assignment phase. Rosters for individual crew members are generated directly from flight leg level. The model presented in this section is similar to the one proposed by Wei et al. (1997) but without generating pairings, considering the special setting in a European airline. The problem is treated as a set partitioning model below, where a set of rosters is given and needs to be assigned to a certain number of the individual crew members, by which all the flights will be covered exactly once. By roster, we mean a slightly different concept to conventional definition of roster in the airline CSP, because a roster here is a shorter line of work for a crew member (with the length of the recovery period). We start with the following definitions prior to the complete model:

(the roster is slightly different from)

F , set of active flight legs (see Section 4.1.3) within the recovery period

F';set of affected flights (see 2.2.1)

D , set of original rosters which represent the previously planned crew schedule for each crew member

R , set of possible rosters

W, set of crew members involved

c_i^w, operational cost of assigning the roster i to the crew member w

u_f, additional cost, if the flight leg f is assigned to a standby or reserve crew member

v_d^w, bonus of assigning an original roster d to its originally assigned crew member w

a_{fi} = 1, if the flight leg f is included in the roster i, 0 otherwise

b_{fd} = 1, if the flight leg f is included in the original roster d, 0 otherwise

e_{iw} = 1, if the roster i belongs to the crew member w, 0 otherwise

The binary decision variables are:

x_i^w = 1, if the roster i is assigned to the crew member w, 0 otherwise

y_f = 1, if flight f is not assigned to a standby or reserve crew, 0 otherwise

z_d^w = 1, if the original roster d is chosen by the crew member w, 0 otherwise

Therefore, the model can be expressed as:

$$\min \sum_{w \in W} \sum_{i \in R} c_i^w x_i^w + \sum_{f \in F} u_f y_f + \sum_{w \in W} \sum_{d \in D} v_d^w z_d^w \tag{4.8}$$

$$\text{s. t.} \sum_{w \in W} \sum_{i \in R} a_{fi} x_i^w + y_f + \sum_{w \in W} \sum_{d \in D} b_{fd} z_d^w = 1 \ \ \forall f \in F \tag{4.9}$$

$$\sum_{i \in R} e_{iw} x_i^w \leq 1 \ \ \forall w \in W \tag{4.10}$$

$$
\begin{aligned}
x_i^w \in \{0,1\} \quad \forall i \in R,\ w \in W \\
y_f \in \{0,1\} \quad \forall f \in F \\
z_d^w \in \{0,1\} \quad \forall d \in D,\ w \in W
\end{aligned} \tag{4.11}
$$

The first two parts of the main objective denote minimizing the total operational cost c_i^w and the additional cost u_f for those flights which are assigned to a standby or reserve crew member. In our approach, we do not explicitly model the assignments to standby/reserve crews, but instead we penalize all those flights that are not assigned to any operating crew. Minimization of the disturbances to the crew is realized by calculating the changes v_d^w in a monetary sense, where a bonus is attached to all rosters that are the same or similar to the originally scheduled rosters. The calculation of the costs will be described later. Constraint (4.9) guarantees that all the flights ($f \in F$) are covered exactly once (more than once would imply the usage of deadhead flights by choosing a set covering type of model), while constraint (4.10) ensures that each crew member ($w \in W$) takes at most one roster ($i \in R$). Obviously, one additional constraint is that all prescheduled activities for each crew member falling in the examined time period have to be covered by their corresponding owner. This constraint is satisfied by building legal possible rosters.

The advantage of such a model can be seen when we are dealing with a disrupted situation where only small changes are applied in the outcome schedule. Due to the strong encourage of choosing originally planned schedules, more crew members will be likely to keep their old schedules instead of getting small changes for everyone. Notably, this advantage can only be true when the disruptions are considerably minor disruptions.

4.2.2 Revised Model

The model presented above has certain drawbacks. Firstly, the variable z_d^w may be possible to be eliminated if we treat every type of roster (newly built and originally planned ones) identically. Secondly, the accuracy is limited because rosters that are “slightly” different from originally planned ones are considered as a completely new built roster rather than a slightly changed plan. Furthermore, the model above cannot easily be adopted by the heuristic approach which will be described in Chapter 5. Due to these reasons, a revised model of (4.12) – (4.16) is proposed below.

Basically, the problem is still treated as a set partitioning model, where a set of rosters is given and needs to be assigned to a certain number of individual crew members, by which all the flights are covered exactly once. Firstly, a set of new or revised notations are listed below:

v_i^w, the penalty to changes (variations) from originally planned schedule
$b_{iw} = 1$, if the roster i belongs to crew member w, 0 otherwise

Therefore, the model can be revised to:

$$\min \sum_{w \in W} \sum_{i \in R} (c_i^w + v_i^w) x_i^w + \sum_{f \in F} u_f y_f \tag{4.12}$$

$$\text{s. t.} \sum_{w \in W} \sum_{i \in R} a_{fi} x_i^w + \ y_f = 1 \ \ \forall f \in F \tag{4.13}$$

$$\sum_{i \in R} b_{iw} x_i^w \leq 1 \ \ \forall w \in W \tag{4.14}$$

$$x_i^w \in \{0, 1\} \ \ \forall i \in R, \ w \in W \tag{4.15}$$

$$y_f \in \{0, 1\} \ \ \forall f \in F \tag{4.16}$$

The first part of the objective function (4.12) denotes minimizing the total operational cost c_i^w, together with the effect of the disturbances to the crew v_i^w realized by expressing the changes in a monetary sense. v_i^w equals zero, if the corresponding roster is identical with an original roster. The calculation of c_i^w and v_i^w can be found in Section 4.1.1. Those flights which cannot be assigned to any crew member in service, require reserve and standby crews, which imposes additional costs u_f. Constraint (4.13) guarantees that all flights ($f \in F$) are covered exactly once, while constraint (4.14) ensures that each crew member ($w \in W$) takes at most one roster ($i \in R$). Such a revised model distinguishes the differences of changes. Consequently, it allows us to manipulate variations in a more flexible way, since every individual change is calculated.

4.3 Model Solving

This section is organized as follows. We first propose a dedicated network structure in Section 4.3.1, through which the enumeration of rosters is realized. Solving the mathematical models as pure integer models is then described: Firstly, the direct solving procedure is introduced in Section 4.3.2; Secondly, a column generation approach is then discussed in Section 4.3.3.

4.3.1 Network Representation

In this section, we embark on building a specialized network in order to execute the roster enumeration and to apply the column generation approach later. Generally, the network is defined in a similar fashion as those proposed by Desrochers and Soumis (1989).

First, a multi-layer network $G' = (N', A')$ is constructed, in which departure and arrival events (nodes) are connected by flight legs (including newly added and updated flights), waiting connections, transits, and hotel stays (arcs) which are operated at multiple home bases (layers). It is an intuitive way to construct such a network initially by representing schedules of crews who station in each home base. That is why we call it multi-layer network. As depicted in Fig. 4.2, each *timeline* is associated with one corresponding airport (including home bases) $p \in P$. All nodes on a timeline are created in chronological order.

The final network $G = (N, A)$, also seen as an acyclic time-space network, is built by eliminating layers and combining timelines that belong to the same airports or home bases (see Fig. 4.3). Additionally, a *source* and a *sink* are created, which can only be visited as the first node and the last node, respectively, i.e., $N = N' \cup \{Source, Sink\}$. By using dummy arcs, source node is connected with each departure node in the network, and sink node is connected with each arrival node. Possible connections between flights can be created between flights when sufficient period of time required for two consecutive flights is found. This can be done by going through all airports, especially home bases, and checking each

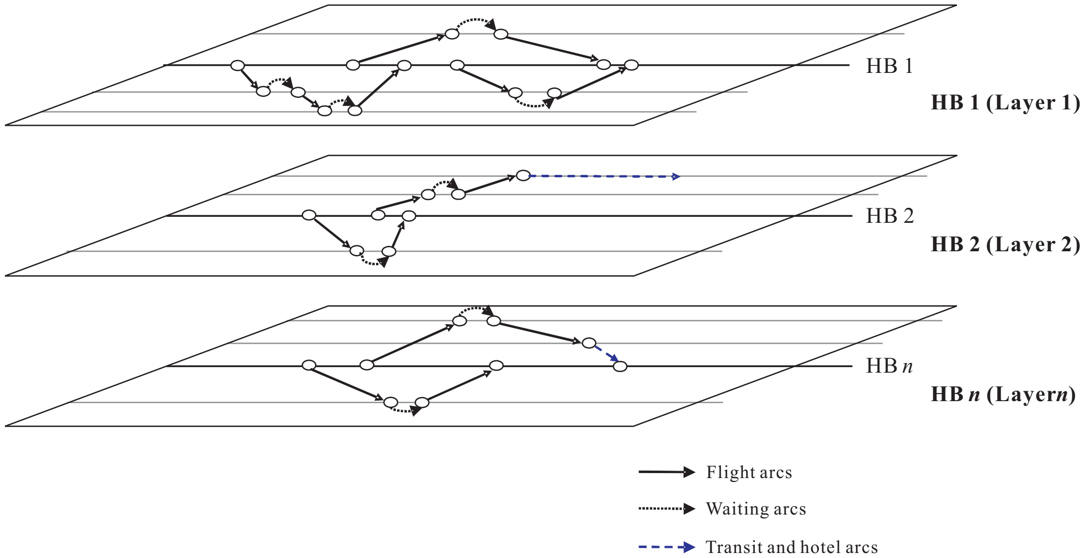


Figure 4.2: Sample multi-layer network G'

pair of arrival node and departure node that may be possibly connected. Additionally, connection arcs are also added between an arrival node and a departure node when it is possible to create a transit in between.

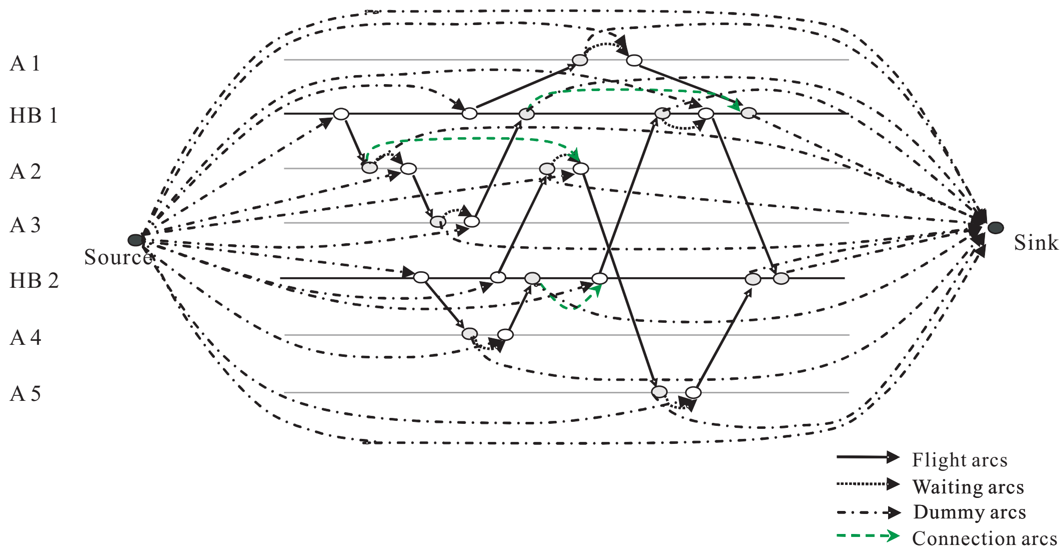


Figure 4.3: Network structure

4.3.2 Problem Solved as an Integer Model

Set partitioning problem is a well-known $\mathcal{NP}$-hard problem (see Garey and Johnson, 1979) which is a problem to find an optimal partition which covers every element exactly once. The examined problem (4.12) – (4.16) is $\mathcal{NP}$-hard, since it can be seen as an instance of set partitioning problem with additional constraints. Similar to a number of practical problems that are examined in many other areas, the airline crew recovery problem formulated as a set partitioning problem has a huge amount of variables. This makes the problem extremely difficult to solve.

The enumeration of all feasible rosters for every crew member is difficult because of the huge number of possible rosters and a number of rules. But it can still be done, at least for small- and medium-sized instances, by enumerating rosters through the network proposed above, since a certain number of basic rules have been considered when building the network, e.g., connections between flight legs. In addition, the recovery period of such a crew recovery problem is usually much shorter than the planning period widely examined in the airline CSP. This thus makes it possible to explicitly create feasible rosters by finding a path in the network. Therefore, apart from exhaustive enumeration method, we also consider enumerate columns by going through all paths in the network in a straightforward way. In summary, all paths are created based on the network, and then translated into rosters for crew members.

In order to find legal rosters for each specific crew member, the network G is first duplicated for each crew member $w \in W$. The resulting network G_w includes all active flights that the crew member may choose within the given recovery period. Based on the current location and the status of the crew member, there are only a limited number of airports that can be taken as starting locations. If the current location is not the departure airport of the first flight, a transit must be provided prior to the flight. The possibilities of creating a transit between two airports are limited due to the airline's policy. Therefore, a subset of airports is selected and considered as possible starting locations. This can be accomplished by eliminating dummy arcs that end with departure nodes starting from undesired airports. Therefore, all paths from the source node to the sink node can likely delineate a possible roster for the crew member examined.

The overall process is a similar procedure to the depth-first-search. Within the search process, a backtracking method is applied to find paths that include nodes with multiple arcs. Each path is then converted to the roster representation, and verified by an isolated module that checks rosters with all relevant rules and regulations. Once a roster is verified and considered as legal, it is added to the roster set R. The above process is repeated for every crew member in the instance. Ultimately, the complete set of all legal rosters for all crew members is created, each of them is then a corresponding variable in the mathematical model presented in Section 4.2.

Once the model is built by enumerating all possible rosters, it can be possibly solved by most standard or commercial IP or MIP optimizers. In our approach we use ILOG CPLEX version 8.0 MIP Solver (ILOG, 2002) to solve the given model. Further discussion will be given in Section 4.4.

When applying the complete enumeration mechanism, small problem instances can be solved very fast. However, it appears to be not practical at all to consider such an approach to solve medium and large instances because of the undesired long solution time. We typically need a long time to enumerate all possible rosters, and the length of time required by CPLEX is also very long. For some instances, the whole process may take several hours to reschedule flights within only a two day period, and CPLEX may take also even hours to solve the resulting model. This drives us to find out a more efficient way to consider such a problem. In the rest of this chapter, a column generation approach will be introduced in detail, which builds the model in an implicit way.

4.3.3 A Column Generation Approach

In general, the airline crew (re)scheduling problem, known as an $\mathcal{NP}$-hard combinatorial optimization problem, usually becomes very hard to solve directly when the problem size increases. In dealing with a practical problem, the direct solving turns out to be inefficient with today's technologies. Even conducting a complete enumeration of all possible rosters can be an extremely difficult task, let alone solving it. Therefore, we have implemented a column generation approach which implicitly builds a promising subset of rosters. Therefore, we may tackle large problems more efficiently. Besides, based on the behavior of such a method we are able to learn the characteristics that the examined instances have.

As mentioned in Chapter 3 for several airline crew pairing problems, column generation methods have been proposed starting with an initial subset of all rosters (columns) as the basis, rather than enumerating a large number of columns explicitly. The master problem (MP) is the LP relaxation of the original IP model. Because the master problem is restricted in a sense of limited number of columns initiated in the basis, it is called restricted master problem (RMP). The roster with negative reduced cost is added into the basis until no more such roster is available. An integer solution is then achieved by embedding the above procedure into a branch-and-bound scheme.

One of the most compelling aspects of column generation is the ability to implicitly create columns that produce the linear optimal solution in the end. Constructing columns in this way allows us to solve large-scale problem instances potentially because we do not have to construct all possible columns as a set beforehand. To demonstrate the procedure, we first describe the basic of column generation method. Given a linear problem, which is formulated as

$$min.\, Z = cx \;:\; Ax \geq b,\, x \geq 0$$

where x and c are decision variables and their cost coefficients respectively, and they are both n vectors. b is a m vector which represents resource assumptions. A is a $m \times n$ matrix, in which each column a_j is associated with a decision variable $x_j\,(j \in J)$. For some problem types, it is impractical or even impossible to build the model with a large number of variables. When solving above linear problem with simplex method, a non-basic variable is priced out in each iteration, entering the basis. The pricing is determined by

$$\text{arg min } \{\bar{c}_j \;:=\; c_j - u^T a_j \,|\, j \in J\}$$

where c_j is the objective function coefficient associated with non-basic variable x_j, and u is the non-negative vector of dual variables.

Since $|J|$ may be huge, a RMP with a reasonably small subset $J' \subseteq J$ of columns is considered. Let $\bar{\lambda}$ and $\bar{u}$ be primal and dual optimal solutions of the

RMP, respectively. When column a_j, $j \in J$ are given as elements of the set A, and the respective cost coefficients c_j can be computed via a function $c : A \to \mathbb{Q}$, the *subproblem*, also called the *column generator* or the *generation problem*, can be formulated as

$$\bar{c}^* := \min \{c(a) - \bar{u}^T a \,|\, a \in A\}$$

Columns priced out then enter the initial subset, and the RMP is solved again. Until there are no further columns whose reduced cost is negative, the problem is solved optimally. In addition, it allows us to deal with complex rules when solving the subproblem, i.e., during the process of columns generation. For a more elaborate description, we refer to Desrosiers and Lübbecke (2003) and Lübbecke and Desrosiers (2004).

The subproblem can be solved typically by: (1) explicit enumeration, (2) constrained shortest path problem by *Dynamic Programming*, (3) resource-constrained shortest path and (4) constraint programming. The method applied for solving individual problems may vary depending on the characteristic of the problem examined. In our approach we model the problem as a constrained shortest path problem which is solved using dynamic programming. Details of the approach will be discussed in Section 4.3.3.3.

Here we outline the column generation method to explain how it works (see Algorithm 1). A set of columns R_{RMP} is constructed by creating an initial set of columns that is able to produce feasible solutions. The restricted problem is solved through function solveLP(), by which the dual vector u is obtained. With the dual information, the subproblem is solved to find a new set of columns R' in which each column $r \in R'$ has the negative reduced cost. The restricted master problem is then extended by adding the columns in set R'. The above procedure is repeated until there is no further column that has negative reduced cost, and the process ends.

For an integer problem (or mixed integer problem), one cannot apply column generation directly as linear programming duality theory is not valid for (M)IPs. Therefore, the LP relaxation of the original integer problem is first created. The column generation is then used to solve the LP relaxation to optimality. But notably, it does not guarantee optimality (sometimes even no feasible integer

Algorithm 1 Column Generation

Require: set of all variables R
Ensure: linear optimal solution to RMP
  $R_{RMP} \Leftarrow$ Initialization()
  assert(RMP is feasible)
  repeat
    $u \Leftarrow$ solveLP(R_{RMP})
    $R' \Leftarrow$ solveSubproblem(u)
    $R_{RMP} \Leftarrow R_{RMP} \cup R'$
  until $R' = \varnothing$

solutions) to the original integer problem, although the optimal solution of its LP relaxation problem is achieved. It is because there might be columns which do not price-out correctly but might be in the optimal integer solution. One way to achieve optimal solution is to add all columns that have a reduced cost smaller than a "gap" value, where gap is defined to be the difference between the optimal linear programming solution and a known feasible integer solution (see Rushmeier et al., 1995). For a range of problems, it is known that the gap between optimal linear programming solution and optimal integer solution of RMP is small enough so that the gap can be acceptable. However, the column generation process may become prohibitive when the gap is large. Another alternative to achieve optimality is applying *branch-and-price*. By applying column generation at each node in the Branch-and-Bound search tree to obtain a bound on IP solution. For a complete review regarding column generation in integer programming, we refer to Wilhelm (2001).

4.3.3.1 Master Problem

In our approach, the master problem can be created by relaxing the integer variables into real variables. Therefore, the constraint 4.15 and 4.16 becomes

$$x_i^w \geq 0 \;\; \forall i \in R, \;\; w \in W \tag{4.17}$$

$$y_f \geq 0 \;\; \forall f \in F \tag{4.18}$$

where x_i^w and y_f are now non-negative real variables. Therefore, the model (4.12) – (4.14), (4.17) and (4.18) represents the LP relaxation of the original integer model which becomes the master problem in the column generation approach. Because only a small part of rosters is initially constructed, the master problem that only includes these initial columns is the restricted master problem of this approach.

The RMP, therefore, is used and solved throughout the column generation procedure interactively. First, it is constructed with an initialization method which attempts to find a small portion of all possible rosters that is able to produce feasible solutions. Once the initial subset of columns is created, dual values that are calculated to price out new columns become available.

The subproblem in our approach is basically a special roster generator which does not aim to find only possible and feasible rosters but rosters that have negative reduced cost. In other words, we seek to construct rosters which can potentially improve the solution value of the RMP. During each iteration of the column generation procedure, new rosters are constructed based on updated dual values.

In the next two section, 4.3.3.2 and 4.3.3.3, we will introduce the initialization methods and specialized subproblem respectively.

4.3.3.2 Initialization

The restricted master problem, as described above, initially includes only a subset of all possible columns. In order to start the column generation, we have to assure that the initial subset provides at least a feasible solution, otherwise dual values cannot be available for pricing out more columns later on. Therefore, a procedure that is able to generate a set of columns and guarantees the feasibility becomes necessary. As proved in Garey and Johnson (1979), finding a feasible solution to a general set partitioning problem is also $\mathcal{NP}$-hard problem. Therefore, the initialization of such a set of columns is a difficult task in general.

In our approach, we have implemented two initialization methods. The first one (see Algorithm 2) is basically a method that creates an initial set of columns little by little. By considering every affected flight, the initialization method tries to create the rosters that contains at least one affected flight. All feasible rosters (set R_{ogl}) that are originally planned are still considered and added into the basis (set R_{RMP}). Because of disruptions, current RMP is definitely infeasible.

Therefore, additional rosters that include affected flights should be generated and should enter the RMP. After locating each departure node and each arc of an affected flight, we create paths in the network G, which represents a number of ($MaxRosterPerAffectedFlight$) feasible anonymous rosters and includes the affected flight. For each crew member, a number of ($MaxRosterPerCM$) feasible rosters are created by examining the feasibility of assigning one of the rosters generated above to the crew member. Finally, the feasibility of the RMP is tested with CPLEX. If CPLEX cannot find any feasible solutions, the value $MaxRosterPerAffectedFlight$ and $MaxRosterPerCM$ are both increased. Then the procedure above is repeated again until the RMP includes feasible solutions.

Algorithm 2 Initialization Procedure 1

Require: affected flights set $F' \neq \varnothing$
Ensure: $RMPisFeasible$ is $true$

```
  MaxRosterPerCM ⇐ n1 − inc1
  MaxRosterPerAffectedFlight ⇐ n2 − inc2
  RMPisFeasible ⇐ false
  initial set of rosters R_RMP ⇐ ∅
  R_RMP ⇐ R_RMP ∪ R_ogl
  while RMPisFeasible is false do
    MaxRosterPerCM ⇐ MaxRosterPerCM + inc1
    MaxRosterPerAffectedFlight ⇐ MaxRosterPerAffectedFlight + inc2
    R′ ⇐ ∅
    for all f ∈ F′ do
      for i = 1 to MaxRosterPerAffectedFlight do
        r′ ⇐ anonymousroster (f)
        R′ ⇐ R′ ∪ {r′}
      end for
    end for
    for all crew member w ∈ W do
      for j = 1 to MaxRosterPerCM do
        r ⇐feasibleRoster(R′)
        R_RMP ⇐ R_RMP ∪ {r}
      end for
    end for
    RMPisFeasible ⇐ feasibility(R_RMP)
  end while
```

Generally speaking, the initialization method described above can effectively generate a limited number of rosters that are able to produce a feasible solution of RMP. It performs well with small- and medium-sized instances. However, it may require a too long period of time to accomplish, as the problem size increases dramatically. Therefore, we improve the above initial heuristic to fulfil the practical demands.

Algorithm 3 Initialization Procedure 2

Require: affected flights set $F' \neq \varnothing$
Ensure: $RMPisFeasible$ is $true$
  $MaxRosterPerCM \Leftarrow n1$
  $MaxRosterPerAffectedFlight \Leftarrow n2$
  initial set of rosters $R_{RMP} \Leftarrow \varnothing$
  $R_{RMP} \Leftarrow R_{RMP} \cup R_{ogl}$
  $RMPisFeasible \Leftarrow false$
  $F_{unc} \Leftarrow F'$
  while $RMPisFeasible$ is $false$ **do**
    $R' \Leftarrow \varnothing$
    for all $f \in F_{unc}$ **do**
      for $i = 1$ to $MaxRosterPerAffectedFlight$ **do**
        $r' \Leftarrow$ anonymousroster (f)
        $R' \Leftarrow R' \cup \{r'\}$
      end for
    end for
    for all crew member $w \in W$ **do**
      for $j = 1$ to $MaxRosterPerCM$ **do**
        $r \Leftarrow$feasibleRoster(R')
        $R_{RMP} \Leftarrow R_{RMP} \cup \{r\}$
      end for
    end for
    $RMPisFeasible \Leftarrow$ feasibility(R_{RMP})
    if $RMPisFeasible$ is $false$ and uncovered flights exist **then**
      $F_{unc} \Leftarrow$ uncovered flights
    else if $RMPisFeasible$ is $false$ and no enough crews **then**
      $F_{unc} \Leftarrow \varnothing$
      $MaxRosterPerCM \Leftarrow inc1$
    end if
  end while

As shown in Algorithm 3, the second initialization method attempts to find more columns by checking which flights are not covered after the restricted master problem is solved every time. In other words, the first method introduced previously is not blindly repeated if the restricted master problem is infeasible. A so called uncovered set (F_{unc}) of affected flights is given, which is initially the same with the complete set of affected flights (F'). Then the first initialization method is used once, and the resulting RMP is solved. If the model is infeasible, we update the uncovered set F_{unc} with only those flights that are not covered and cause the infeasibility, reported by CPLEX. With such a considerably small number of uncovered flights, the above procedure is repeated until the RMP is feasible.

In case of infeasibility, another possible reason is that there are not enough crew members. In other words, some crew members have to take more than one roster in order to assign all flights entirely. In this case, more rosters will be generated for each crew member by changing the value of $MaxRosterPerCM$. The procedure is then repeated until a feasible solution can be found. In our approach, we observed that such case of infeasibility only happened in a few tests, but the majority of cases is that some flights are not uncovered.

In our approach, we first set the values of both $MaxRosterPerAffectedFlight$ and $MaxRosterPerCM$ to 1 in favor of possible reduction of the complete time. If the resulting RMP after the first iteration is infeasible and the reason of infeasibility is uncovered flights, both values are increased by 2. For medium- and large-sized instances, the value is set comparatively larger, e.g., 4 or 6. If the infeasibility is caused by the limited number of crew members, the same number of rosters is generated repeatedly for every crew member until the problem becomes feasible.

4.3.3.3 Subproblem

Solving the RMP presented in Section 4.3.3.1 yields the dual multipliers u_f and u_w for constraints (4.13) and (4.14), respectively. Accordingly, the reduced cost for each column (variable) x_i^w can be expressed as

$$\bar{c}_i^w = (c_i^w + v_i^w) - {u_f}^T a_i - {u_w}^T b_i \tag{4.19}$$

Therefore, the subproblem examined here is to find columns which have negative reduced cost calculated above. In our approach, an enumeration method and a constrained shortest path algorithm are implemented and tested to achieve the task of rosters generation.

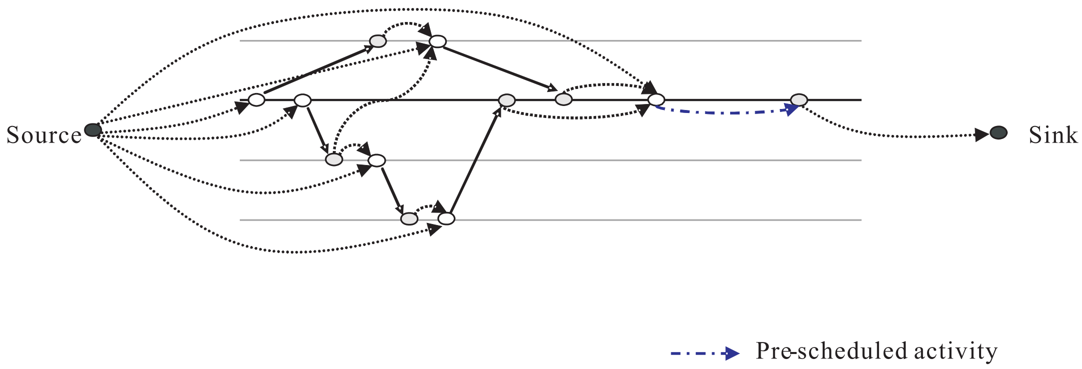


Figure 4.4: The reduction of the network

First we consider the possible reduction of the network by pruning nodes and arcs that are unnecessary for specific crew members. The significant reduction can be achieved when pre-scheduled activities of a crew member are incorporated into the network as arcs. Because the arcs representing pre-scheduled activities must be passed by any valid paths, all arcs and nodes that lie within the periods of them can be seen as unnecessary ones therefore can be eliminated from the network. As illustrated by the example in Fig. 4.4, a crew member has been assigned one pre-scheduled activity that takes place at the end of the recovery period. Therefore, those nodes and arcs that conflict with the pre-scheduled activity in terms of time can be safely removed from the network. Furthermore, only the arrival node of such a pre-scheduled activity arc is connected to the sink node, because this arc must be taken and there is no arrival node available after the pre-scheduled activity arc. Notably, such arcs may appear in the middle of the recovery period. Accordingly it is possible to eliminate not only the nodes and arcs that have conflicts in time, but also the arcs from the source node to all departure nodes after the pre-scheduled activity and the arcs from all arrival nodes before the pre-scheduled activity to the sink node.

Prior to the implementation of the constrained shortest path algorithm, we make a preliminary experiment using an enumeration method to generate columns that have negative reduced costs. Basically, it is a comparably inefficient way to find eligible columns, but it is still faster than the pure enumeration method introduced previously. All valid paths are created by starting from the source node to the sink node in the network. The legality of each path is examined by considering rules, e.g., flight duty time limit, and the cost is calculated if the corresponding roster is legal. All remaining rosters are then sorted by their costs. That is to say, rosters with no transit and hotel stay will stay on the top of others. The reduced cost of each roster is calculated, and rosters with negative reduced cost are kept. Once the set of rosters with negative reduced cost is built, a certain number of such rosters are added to form a new restricted master problem.

Before the description of the constrained shortest path algorithm, we define the cost $\hat{c}_{ij}^w$ for each arc of the network. By incorporating the dual values u_f, we calculate such a cost as

$$\hat{c}_{ij}^w = c_{ij}^w + v_{ij}^w - \sum_{f \in F_{ij}} u_f \qquad i, j \in A \tag{4.20}$$

Where c_{ij}^w and v_{ij}^w denote the operational cost and the change cost of the arc (i, j), respectively, Note that the flight arc does not impose any operational cost but possibly change cost. In contrast, those arcs that are translated into transits and hotel stays incur operational cost (see Section 4.1.1) but no change cost is introduced.

Further restrictions on paths can be modeled by attaching the associated vector of resources consumption $RC_a = (RC_{a1}, RC_{a2}, ..., RC_{aq})$ to each arc $a \in A$ from node i to j, where q is the number of resources examined in the approach. In our approach, flight duty time and weekly rest restrictions are handled as resources. The quantities of each resource consumption for a path from source to the node j can be calculated by adding the corresponding resources consumption that are accumulated from the source until node i. Therefore, the legality of the path can be checked by examining the associated lower and upper bound for each resource. We define the label for a path as the following vector:

$$L = \begin{pmatrix} \hat{C} \\ RC \end{pmatrix} \tag{4.21}$$

Consider two paths a and b ending at node i, a is dominated by b if $L_a \geqslant L_b$. This dominance process can also be understood as an elimination procedure, in which unpromising paths are discarded. In other words, at a given node i, any illegal and dominated path from the source node to the node i is eliminated.

Start with the source node, all nodes are visited by flowing along arcs. At given node i, all valid and nondominated paths from the source to the node i are stored and compared. Finally, at the sink node paths with negative cost are added into the master problem. After the updated restricted master problem is solved, the new dual information is obtained and the procedure above is repeated until no further paths with negative reduced cost exist. For more elaborate description of such an algorithm, we refer to Desrosiers et al. (1995).

4.4 Computational Experiences

In our approach, we carry out a number of experiments based on the data from a European airline. The characteristics of the problem instances have been discussed in Section 2.5. Numerous test runs have been conducted on a regular PC with Intel Pentium IV 2.2 GHz CPU, 2 GB main memory, running Microsoft Windows XP Professional operating system.

The applied optimizers were ILOG CPLEX 8.0 (ILOG, 2002). We used the default parameter settings for CPLEX, except that we set the parameter Probe to 1, indicating a higher probing level slightly above default. In doing this, the solution time may decrease slightly. After numerous experiments, we observed that invoking aggressive cut strategies generated a large number of cuts, but also increased the solution time significantly. Therefore, it is not worth generating cuts aggressively, since it slows down the solving process in most cases.

Table 4.1 lists a collection of practical problem instances we tested in our approach. The table provides an overview of these instances in terms of the numbers of home bases, airports, flights, crew members etc.

Table 4.1: Overview of the problem instances

Instances	Home bases	Hotel bases	Airports	Total flights	Crew members	Duration (day)
A1	6	8	52	159	47	15
A2	5	10	57	228	42	15
A3	6	13	66	268	42	10
A4	6	14	65	275	42	10
A5	6	14	66	406	42	15
A6	6	16	69	415	42	15
B1	11	21	66	1,287	188	10

Table 4.2: Disruption scenarios based on instances A3 and B1

Disruption scenario	Inst.	Disruption summary		Recovery period (day)	Active flights
		Unavailable crew members	Affected flights		
A3-1-1	A3	0	2	1	27
A3-1-2	A3	2	2	1	27
A3-2-1	A3	4	2	2	53
B1-1-1	B1	1	0	1	117
B1-2-1	B1	2	4	2	242
B1-2-2	B1	3	4	2	242
B1-2-3	B1	1	6	2	242
B1-2-4	B1	2	4	2	242

Various artificial disruption scenarios are created based on available instances. Scenarios proposed are usually one or two days recovery period, within which a certain number of flights are affected (delayed, cancelled or newly added) and several crew members are not available. Out of the variety of available problem instances we present a selection within the following table 4.2.

For column generation approach, the number of columns generated during each iteration also affect the total number of iterations needed. As one can see from Fig. 4.5, the more columns we generate in each iteration, the less number of iterations we need to complete the overall solving process.

In addition, the number of columns generated during each iteration may affect the total solution time of the enumeration based column generation (see Fig. 4.6). As illustrated by the figure, a value around 40 appears to be the best number of

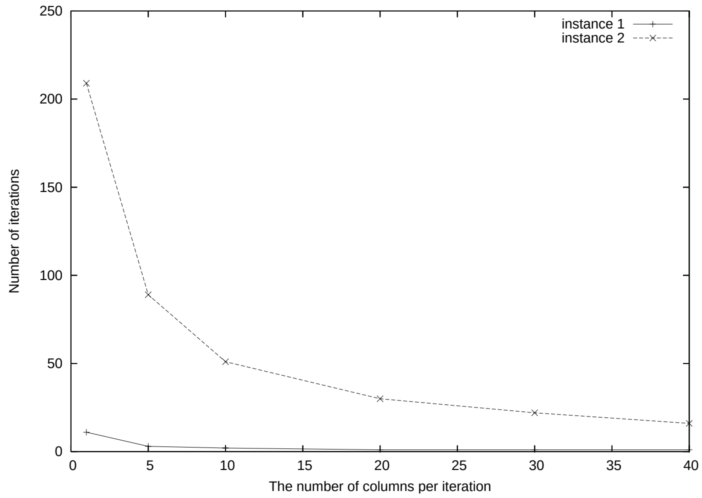


Figure 4.5: The effect of the number of columns generated during each iteration to the number of iterations

columns during each iteration.

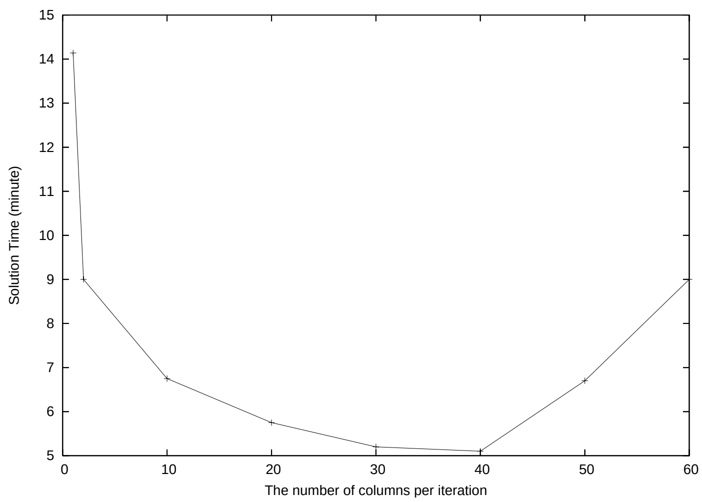


Figure 4.6: The effect of the number of columns generated during each iteration to the total solution time

In Table 4.3, a comparison among the different approaches is presented. In these tests, we simplify the calculation of the change cost v_i^w by associating a negative constant change cost to rosters that are the same as originally planned

Table 4.3: Computational performance between enumeration based approach and column generation approach

Disruption	Direct solving		Col-gen (enumeration)[a]			Col-gen (CSPP)		
scenario	time[b]	obj.	iter.	time	obj.	iter.	time	obj.
A3-1-1	10	100%	2	12	100%	2	7	100%
A3-1-2	9	100%	2	9	100%	2	7	100%
A3-1-2	15	100%	3	11	100%	3	8	100%
B1-1-1	51	100%	2	32	100%	2	21	100%
B1-2-1	3643	100%	8	235	100%	12	61	100%
B1-2-2	3719	100%	7	229	100%	10	59	100%
B1-2-3	3560	100%	7	226	100%	10	47	100%
B1-2-4	3287	100%	7	220	100%	12	35	100%

[a] 40 columns per iteration
[b] Computational time in seconds

ones. The negative value used is the same as the bonus value applied in the basic model. Both column generation based approaches provide faster solution times than the direct solving approach. Especially for large instances, the column generation approaches can reduce the total solution time significantly, because the direct solving method requires a tedious enumeration of all possible rosters. The direct solving approach needs not only a longer time for solving the huge model generated, but also a longer time to generate all possible rosters and to calculate their costs. Interestingly, both column generation approaches can produce an optimal solution in all cases presented here, and it sometimes is not identical to the solution found by the direct solving approach. The reason that the column generation method can possibly find an optimal solution in our approach is because there exist multiple optimal solutions which introduce the same cost but with different assignments.

4.5 Summary

In this chapter, we model the airline crew recovery as set partitioning problems. A column generation method is proposed to solve the problem in an implicit way. Its performance is compared with a direct solving approach, which shows a significant improvement in terms of the required computational time. Mostly, the column generation method can find optimal solution although it is not guaranteed theoretically. After conducting the comparison between the two methods for solving the subproblem, we observe that constrained shortest path algorithm can solve the subproblem much faster than the enumeration based one. Therefore, as shown by the computational results, the column generation shows the greater potential to find an optimal recovery solution.

However, certain problems also arise as it is applied to deal with large problem instances. A large number of daily flights and crew members and a long recovery period all have dramatic influence on the general recovery performance significantly. In such a case, the proposed method may require longer computational time to find a final solution, which may exceed the limit of desired recovery time. Therefore, it is necessary to develop a solution method which can find an acceptable and applicable solution within a reasonable period of time. This practical observation therefore motivates us to develop heuristics which will be introduced in Chapter 5.

Chapter 5

Heuristics for Airline Crew Recovery

For several decades a variety of heuristical methods have been proposed to solve combinatorial optimization problems and become more and more popular recently, particularly for some real life problems that seem to be extremely difficult to solve. Due to their simplicity and rapid problem solving characters, various categories of heuristic algorithms have been chosen to deal with a number of practical problems, e.g., planning and scheduling problems, where they are able to produce good or even optimal solutions within a reasonable amount of time. Except for those heuristics that are solely dedicated to a specific problem, heuristics can be classified into several groups, such as constructive heuristics, local search based heuristics, evolutionary algorithms etc.

Over the last 15 years much of the research effort has been concentrated on the development of *metaheuristics*, using mainly two principles: *local search* and *population search* (see Hertz and Widmer, 2003, for an overview). Basically, local search based methods perform the intensive exploration of the solution space by moving at each step from the current solution to another promising solution in its neighborhood, such as simulated annealing, tabu search and variable neighborhood search etc. Differently, population search consists of maintaining a pool of solutions and recombining them in order to hopefully produce better solutions, such as genetic algorithm (GA) and adaptive memory procedures etc. In many papers it is approved by many researchers that these techniques show great efficiency in solving “hard” problems.

As one may see in many scientific publications, genetic algorithms have been successfully applied to solve many problems, as they can gradually find better solutions during the course of an evolution. Lots of research has been carried out after the concept of genetic algorithm was first introduced by Holland (1975). For example, the set covering and partitioning problem, as classical combinatorial problems, were systematically examined by Beasley and Chu (1996) and Chu and Beasley (1995) respectively, who observed the potential for solving these types of problems. Although genetic algorithms may not be computationally competitive for every problem, Chu and Beasley have observed and envisaged that for solving set partitioning problems their GA will become more effective than CPLEX either when the problem is very big or when there is a considerable gap between the LP relaxation solution value and the optimal integer solution value. More specific applications can be found in e.g., Alcaraz and Maroto (2001), Dias et al. (2001), Fu et al. (2003), Wall (1996) and Xu and Louis (1996).

As an analogy to the theory of evolution in biology, a genetic algorithm basically works with a group of candidate solutions (*individuals or chromosomes*), called *population*. Each individual in the population is encoded into a specific representation with regard to the problem. The new generation (*offspring*) is produced from one or more individuals (*parents*) by applying recombination methods called variation operators, e.g., *crossover* and *mutation*. Every individual is measured and attached a value (*fitness value*) showing how "good" it is. The selection of parents and the survival of offspring may be determined randomly or based on their fitness value. In this way, the convergence to an acceptable or optimal solution is accomplished. The outline of a general GA is given in Algorithm 4.

One reason that genetic algorithms are an interesting solution approach for the airline CRP may be the following: (1) GA is very flexible by applying various operators and examining a number of parameters; (2) GA shows the implicit parallelism and the "intelligent" probabilistic search; (3) GA can be extended by incorporating other search methods, e.g., local search based heuristics; (4) Similar approaches have been widely studied, e.g., GA for generalized and specific set partitioning/covering problems.

The emphasis of this chapter lies on heuristic based methods for solving the airline crew recovery problem. We begin with a dedicated genetic algorithm in

Algorithm 4 Outline of Genetic Algorithm

$g \Leftarrow 0$
$p \Leftarrow$ initPopulation()
evaluate(P)
repeat
  $\{p1, p2\} \Leftarrow$ selection(P)
  $p' \Leftarrow$ crossover($p1, p2$)
  mutate(p')
  evaluate(p')
  survive(p', P)
  $g \Leftarrow g + 1$
until terminating condition

Section 5.1, and then introduce a constructive heuristic in Section 5.2.

5.1 A Genetic Algorithm for the Airline Crew Recovery Problem

In this approach, we customize the conventional genetic algorithm into a hybrid genetic algorithm. It includes a set of heuristics with the knowledge of the nature and domain of this specific problem, together with a so-called *local improvement* procedure acting as a supplementary local search to the genetic algorithm.

To our knowledge, a pure genetic algorithm is not able to perform well if the problem centered methods are not implemented and incorporated. Throughout this section, one can observe that the knowledge of this particular problem, namely the airline crew recovery problem, influences every part of the algorithm.

In this section, we will solely elaborate on a genetic algorithm based approach to solve the airline crew recovery problem. Starting with a dedicated two dimensional representation (Section 5.1.1), we describe the initialization of the population in Section 5.1.2, the application of various operators in Section 5.1.3, the evaluation of individuals in Section 5.1.4, the feasibility maintenance in Section 5.1.5, and selection and replace scheme in Section 5.1.6 and 5.1.7, respectively. Finally, computational results tested on data from a medium-sized European airline are given in the end of this section.

5.1.1 Two Dimensional Representation

Unlike other previous attempts for the airline crew (re)scheduling problem, we apply an intuitive matrix based two dimensional representation, which is inspired naturally by the two dimensional structure of airline crew schedules.

5.1.1.1 Matrix Encoding Scheme

The most natural representation of a chromosome in context of GA is probably a one-dimensional string type of form, such as the ordered city list for the traveling salesman problem (see Michalewicz and Fogel, 2000, chap. 7) and the activity permutation list for the resource-constrained project scheduling (see Hartmann, 1998). However, according to our experience, string representation is not suitable to solve the airline crew scheduling and recovery problems because of the complex structure of their solutions. Most previous attempts adopt one dimensional string representation as the encoding scheme, e.g., El Moudani et al. (2001) apply a non-binary string representation, in which the ith component of the chromosome indicates that the corresponding crew member is assigned to the ith pairing. In other words, such an encoding scheme requires a set of pairings created prior to the GA. Another disadvantage is the lack of possibility of manipulating pairings, since the result of such approach is the final crew schedules that are personalized and assigned to individual airline crew members.

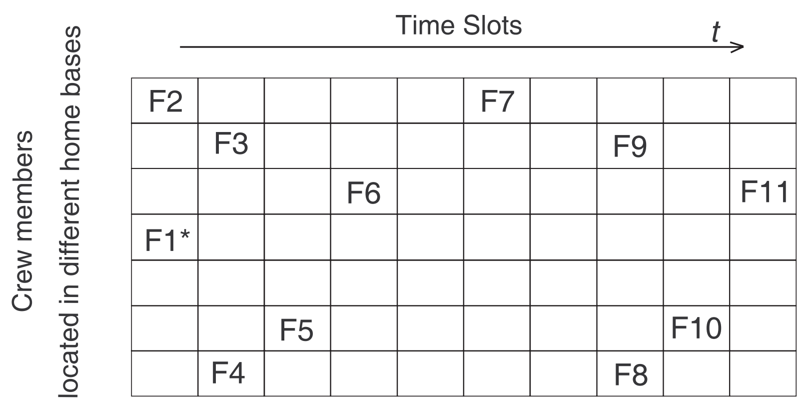


Figure 5.1: Two-dimensional representation

However, due to the specific problem structure we propose a two-dimensional matrix based encoding scheme which represents the direct assignment of flight legs to crew members. One example chromosome can be seen in Fig. 5.1. As depicted in the figure, each column of the matrix indicates a certain time slot (e.g., each time slot represents one hour time span), within which a set of flights depart. Each row stands for a roster of an individual crew member who is associated with the current row, i.e., the number of rows is exactly the same as the total number of crew members examined. All elements of the matrix represent the sequential numbers of the flight legs, which is unique for every problem instance. In addition, pre-scheduled activities, such as vacancies, simulators and requested off-duty etc., are represented as special flight legs with the additional indication that they are fixed and hence cannot be changed and/or removed from the given cell (marked with an asterisk, e.g., the flight leg F1 in the first column in Fig. 5.1).

The benefit of such an encoding scheme can be seen as follows. Firstly, pairings are not involved in the approach, i.e., we directly deal with flight legs instead. This makes it possible to construct better solutions or achieve a global optimal. Secondly, the matrix based representation eases the implementation of various operators introduced later, as a one-dimensional string representation cannot model the complex solution structure. It becomes more difficult for the string type of encoding if there are multiple home bases available in the given problem.

5.1.1.2 Constraints Consideration in the Matrix Encoding

By adopting the concept of time slots in the matrix, we are able to avoid possible violations of some constraints on forming a legal crew schedule. Every change in the matrix thus directly reflects the change in the crew schedule. Due to such a specialized representation, changes to the crew schedule underlie the following assumptions:

- Flight legs can only be moved between rows but cannot be moved to different columns as the departure times are fixed to exactly one time slot (column).

- One cell in the matrix can only contain at most one flight leg, because no pair of two sequential flight legs depart within the same time slot. (it is impossible to service two flights in less than one hour)

- A flight leg can only be assigned to one crew member, as only one crew position is involved at a time, i.e., each flight leg is unique within one matrix.

5.1.2 Population Initialization

Initialization is a process to generate the initial population which allows the application of variation operators. A certain number (population size) of individuals, exhibiting equal or similar genome structures, is created either randomly or heuristically. Generally speaking, seeding with a randomly generated solution in the initial population comes along with wide diversity. However, some researchers have reported that a population with a higher quality starting solutions obtained from dedicated procedures can help the algorithm perform faster and find better solutions (see Reeves, 1993, chap. 4). As a result, the risk of premature convergence also increases.

The random initialization of the starting population is done within two steps. First, it assigns the pre-scheduled activities to their corresponding crew members, such as vacation days, flight simulators or other requested-off. Because these activities cannot be assigned blindly and changed by mutation or crossover operators, they have to stay in their predefined cells. In the second step, the flight legs are randomly assigned to the crew members by simply putting them to one of the rows in the matrix. Notably, flight legs are only placed in the matched columns of their corresponding time slot.

In this approach, apart from the random initialization of the population, two other strategies are applied to generate initial individuals: The first initialization heuristic checks the arrival time of the previous flight leg, every time a flight leg is assigned. If the flight leg does not fit in a specific position of the matrix, another crew member needs to be evaluated. If no further row is found, the flight leg is assigned randomly. By using this heuristic, the individuals start with a higher fitness value, although most of them may be still infeasible. However, the drawback is that individuals sometimes get very similar and the risk of ending up in local optima increases.

Algorithm 5 Evolutive Initialization

Require: affected flights set $F' \neq \varnothing$
Ensure: $F' = \varnothing$
  $F \Leftarrow F - F'$
  for all $f \in F'$ **do**
    while f is not assigned **do**
      $w \Leftarrow$ random $(w \mid w \in W)$
      if feasible assignment(f,w) is true **then**
        assign (f,w)
        $F' \Leftarrow F' - \{f\}$
      end if
    end while
  end for

Here one can note the fact that one part of the objective function is to minimize the changes from the original schedule, i.e., parts of the final solution stay unchanged. Therefore, we take the advantage of such characteristics of the problem, a procedure, *evolutive initialization*, is implemented. It generates a certain number of individuals by keeping unaffected flights F_{unaff} ($F_{unaff} \subset F$) intact and assigning the rest to crew members who are able to provide the service (see Algorithm 5). In other words, most crew members do not change their schedules, if they are not directly influenced by current disruptions.

According to our experience, in order to apply an effective genetic algorithm, an initial population with a high diversification is very useful and seeding individuals also fasten the convergence. Therefore, it can be achieved by combining the three methods described above, i.e., random, heuristic and evolutive initialization procedure. In this approach, we create 20 percent of individuals by the random initialization method and 1 individual using evolutive initialization method, and the rest is generated through the heuristical procedure described previously.

5.1.3 Variation Operators

For the string representation based genetic algorithm, two commonly applied operators are two-point crossover and single bit mutation. However, specific operators have to be designed for this real-life complex problem. In this approach, two main categories of variation operators were applied and tested. The first category involves the crossover type operators which select two individuals as parents and recombine them; the second category includes the mutation type operators which are applied on only one individual.

5.1.3.1 Crossovers

We started with implementing a simple crossover operator, *row-based crossover*, in which a given range of rows of one parent are replaced by corresponding rows from the other parent (see Fig. 5.2). However, as a consequence a flight leg is often served by more than one crew member in the offspring, or a flight leg is not included at all. Therefore, we propose a correction procedure to deal with both effects by inserting missing flight legs randomly, and deleting doubled flight legs randomly.

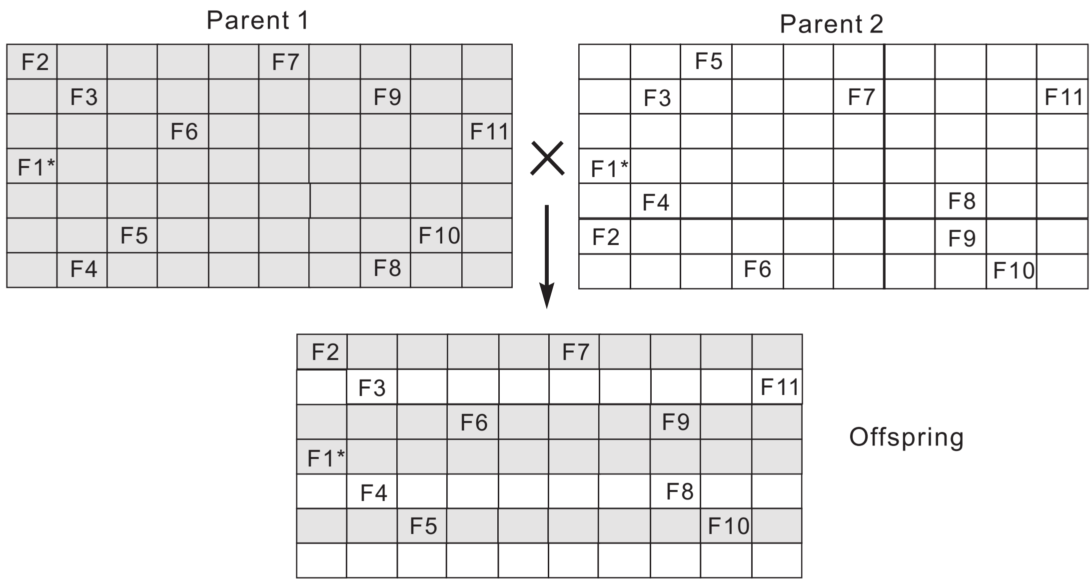


Figure 5.2: *Row-based crossover* operator

In contrast to the row-based crossover, a so-called *column-based crossover* was designed. Its basic idea is to construct new offspring in a way that columns are selected randomly or heuristically from the parents chosen. As illustrated in Fig.

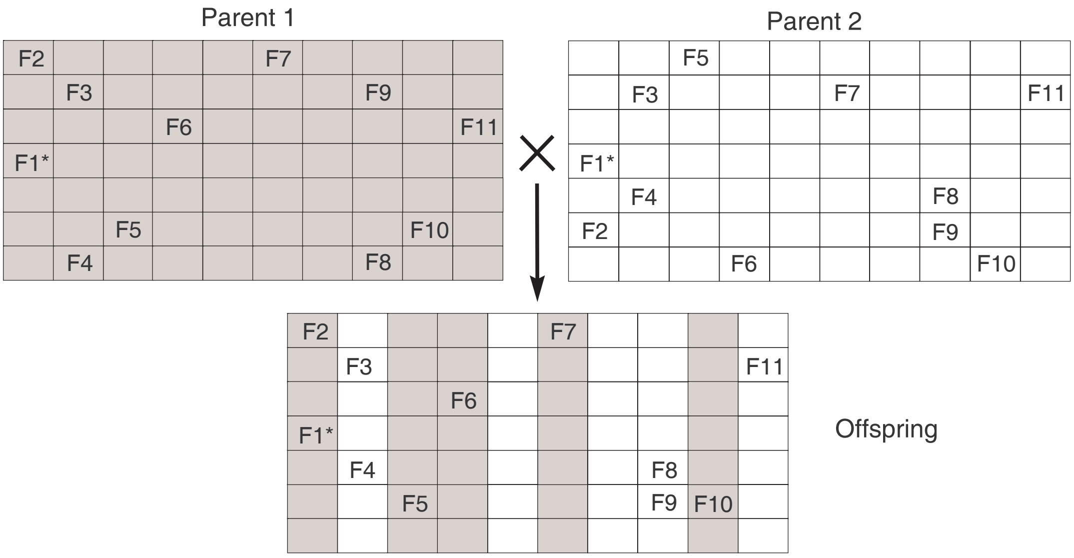


Figure 5.3: *Column-based crossover* operator

5.3, in order to maintain the feasibility of the offspring, this special crossover tries to preserve "good" ranges of the columns containing valid pairings. In other words, feasibilities are likely preserved, when two consecutive flight legs are connected at the same airport, and there is sufficient time between the two flight legs. However, it is still possible to have infeasible sequence of flights assigned to a crew member. For example, in the figure one can see that the sequence of F9 and F10 can express infeasibility, because the departure time of F10 is earlier than the arrival time of F9, or the ground time between the two flights is not adequate.

5.1.3.2 Mutations

The intention of mutation operators is to avoid getting trapped in local optima through increasing the diversity of the population. It usually modifies the resulting offspring slightly by changing only few values or varying a part of the individual. Here we introduce two mutation operators which have been applied in our approach.

First, a so-called *basic mutation* chooses randomly one flight leg f from a chromosome to be mutated and one candidate crew member in the crew member set W. If the position of the chosen destination row is empty, it is moved to that position (see Fig. 5.4.a). If not, these two flight legs are swapped (see Fig. 5.4.b). Nothing is changed, if the new position is exactly the same to the current position. By doing this, we are able to slightly modify an individual and generate a new one.

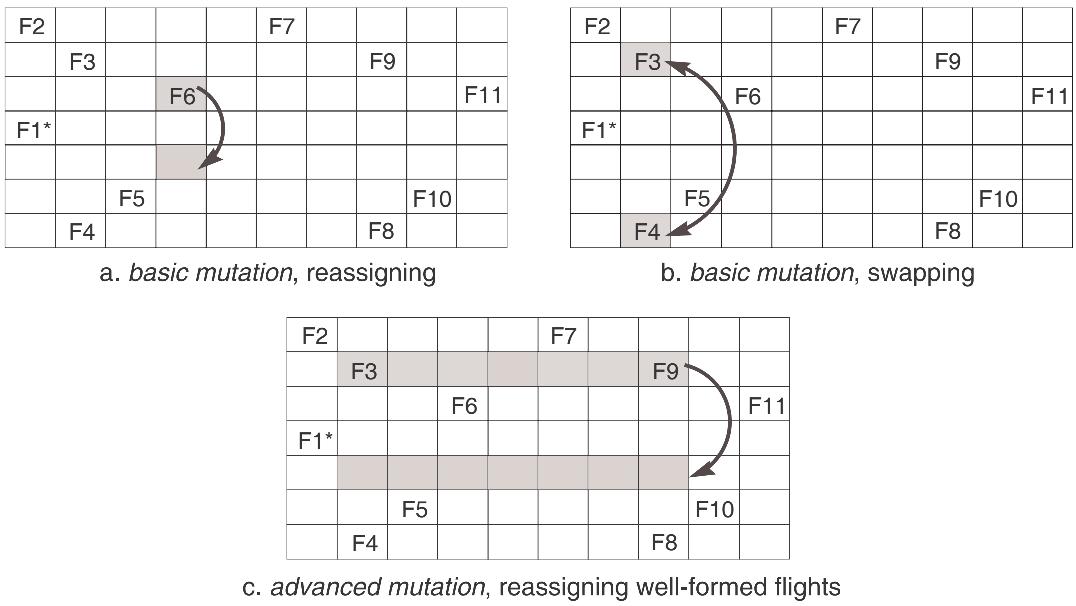


Figure 5.4: Mutation operators

The second mutation operator, *advanced mutation*, includes a heuristic procedure that substitutes the random selection of the new crew member described above. After a flight leg is randomly chosen, a candidate crew member (one row in the matrix) is selected and evaluated according to certain restrictions: We take into account whether the possible transit exists and whether subsequent flight legs are suitable for the new inserted flight leg with regard to their departure and arrival airport and the underlying times. For all crew members satisfying the above restrictions, the corresponding cost is calculated for assigning the flight leg (the detail of the cost calculation can be found in Subsection 5.1.4). The crew member with the lowest cost is, therefore, chosen, and the flight leg is then assigned to this crew member. Similarly, if some flights already occupy the given position, then these two flight legs are swapped. Furthermore, a complete sequence of flight legs is also moved to the new row, if the sequence of flights can be recognized as "well-formed" (normally but not necessarily one day work for a crew member, see Fig. 5.4.c as an example). Such a sequence of flights can be identified by checking arrival/departure airports, as well as arrival/departure times of the flight legs.

5.1.4 Evaluation

According to the nature of the evolution, one has to find out which individual may survive in what sense of the measurement. The fitness of an individual is determined through a proper evaluation function. Normally, each individual's fitness value calculated by the evaluation procedure is a number expressing the quality of the solution.

Generally speaking, one common way to calculated fitness value of an individual is to combine its cost and the penalty. The cost of an individual can be easily translated from the objective function of the given problem. However, it does not take infeasibility into account and individuals in the population are likely infeasible solutions, hence an appropriate penalty has to be incorporated into the evaluation process. Accordingly, the fitness function $f(x)$ can be of the form

$$f(x) = c(x) + p(x) \tag{5.1}$$

where $c(x)$ is the objective function value (cost) and $p(x)$ is a penalty method which is usually problem specific. In this section, we attempt to elaborate on the calculation of the cost and the penalty which are associated to each individual.

Basically, the evaluation procedure is applied to each individual, determining its quality compared to the whole population. In our approach, three "costs", *real cost*, *virtual cost* and *change cost*, are associated with each individual. As the objective is to minimize the cost calculated by the objective function introduced in Chapter 4, the real and change costs constitute the cost function $c(x)$. $p(x)$ then becomes the function that calculates the virtual cost associated with each individual.

The real cost C_{opl} (operational costs) is the incurred cost in the corresponding solution that is translated from the individual. It consists of hotel stay cost that arises when a crew member takes an overnight rest at another airport rather than his or her home base, and also the transit costs that are imposed when crew members are transferred from one airport to another by means of taxi or train. Detailed description is already given in Section 4.1.1.1 of the last chapter.

The virtual costs C_{inf} have been implemented as a penalty for those individuals that do not comply with restrictions, i.e., they are infeasible solutions. For example, in the case that it is impossible to create a feasible connection between two flight legs, a high penalty for the connection is introduced. Likewise, a certain penalty is also imposed, if the time between two flight legs is too short for the crew member to check-out and check-in. Obviously, a penalty must be given if the next flight starts even earlier than the arrival time of the previous flight. Because an effective penalty method should be strong enough so that the GA may not search only among infeasible individuals (see Richardson et al., 1989). Therefore, the penalty method applied in our approach is of "strong", since constraints that are violated by infeasible solutions are considerably "hard" constraints to the given problem.

In addition to these main restrictions, there are other restrictions which the algorithm has to take into account. These rules and regulations are normally defined by the airline, collective labor agreements and civil aviation authorities. For example, the violation of the following rules may incur virtual costs:

- Maximum daily/weekly/monthly flight hours
- Minimum rest time between flight duties
- Maximum sequential working days and minimum weekly rest days
- Weekly rest at crew member's home base

These rules are checked by examining each row of the matrix. For example, the flight time is accumulated, and a penalty is imposed whenever it exceeds the given upper bound of the total flight time within a day. More details can be found in Section 2.2.3.

The change cost C_{chg} is the sum of change penalty v_i^w of all the rosters in the solution. It is calculated in a way that a certain amount of penalty is set to each change depending on how much the flight is involved in the situation and which category the change belongs to. (4.4) and (4.4) show the calculation of such change cost.

5.1.5 Feasibility Maintenance

Although the traditional genetic algorithm works well on some particular problems, hybrid genetic algorithms, especially in combination with algorithms reflecting specific knowledge of the examined problem, usually perform better. As stated in Levine (1996), many researchers have provided mounting experimental evidence showing that hybridizing a GA with a local search heuristic is beneficial.

Algorithm 6 Local Improvement Procedure

Require: individual s
Ensure: feasible solution s
  for $i = 1$ to ROW **do**
    for $j = 1$ to COL **do**
      if $s[i][j] \neq \varnothing$ and $s[i][j]$ is not pre-scheduled **then**
        $f \Leftarrow s[i][j]$
        if checkFeasibility(f, s) is $false$ **then**
          $s[i][j] \Leftarrow \varnothing$
          $\{w_1, w_2, ..., w_q\} \Leftarrow$ availableCM(f, W, s)
          if $q > 0$ **then**
            $w \Leftarrow$ random$(\{w_1, w_2, ..., w_q\})$
            assign(f, w, s)
          else
            $w' \Leftarrow$ findPotentialCM(f, W, s)
            swap(f, f', w, w')
          end if
          evaluate(s)
        end if
      end if
    end for
  end for

As some of the operators presented above might produce an infeasible solution, a dedicated local search procedure, called *local improvement*, is applied under a specified probability right after the generation of new offspring. Strictly speaking, the purpose of such a procedure is to improve the solution by finding one of its feasible neighbor solutions.

Basically, this procedure goes through all infeasible assignments in the solution, and repairs them by reassigning the flight legs to other crew members (see Algorithm 6). A flight leg $f \in F$ with infeasible assignment is assigned to one of randomly selected crew members that is found by function availableCM(f, W, s). Feasibility is maintained by adding a filter into the function to select only those crew members who are approved to be suitable to take the flight, or by swapping flights between crew members which will produce a feasible solution and possibly improve the current solution. The reason why we choose such a straightforward heuristic method rather than other well-known modern heuristics (e.g., simulated annealing), is that the running time for those type of local searches is usually very long therefore not acceptable in practice.

5.1.6 Selection Method

The selection of parents is the process that provides a change of reproduction to every individual in the population. As described earlier, some types of variation operators, such as crossovers, require two or more parents to produce new offspring. The quality of the resulting offspring may depend on its parents because most parts of the offspring are inherited from them. A number of selection methods are devised and applied, such as random selection, proportionate selection (roulette-wheel), fitness scaling, tournament selection, truncation selection and ranking-based selection. Because the detailed introduction of different selection methods is beyond the scope of this thesis, hence we refer to Bölte and Thonemann (1996), Michalewicz and Fogel (2000) and Thierens and Goldberg (1994) for more details.

In our approach, in order to guarantee an appropriate selection and reproduction, a ranking-based selection scheme is considered. Because the variance among fitness values can be quite large, the risk that some individuals dominate the whole population after a few generations must be taken into account. Hence, ranking-based selection strategies rather than strategies based on the absolute fitness value seem to be more applicable for this specific problem. The possibility that an individual is selected to the reproduction is determined by

$$P(i) = \frac{R(i)^s}{\sum_{j=1}^{N} R(j)^s} \qquad i \in \{1, ..., N\} \tag{5.2}$$

where $R(i)$ is the rank of the individual i according to its fitness value. Because the examined airline crew recovery problem is to find the solutions that have minimal costs (fitness value), the algorithm intends to find the individuals with lowest fitness value. In other words, the lower the fitness value is the better the individual is. Therefore, the individual ranked as the last position represents the best one in the population. N equals the total number of individuals in the population, also the ranking of the best individual in the population. The sum $\sum_{j=1}^{N} R(j)^s$ normalizes the probabilities to ensure that $\sum_{i=1}^{N} P(i) = 1$. With the exponent s, it is possible to produce the probabilities according to the rankings more significantly. If $s > 1$, the differences between two probabilities increases. In doing this, better individuals have not only a much higher chance to be selected. After a number of tests, setting s to 1 mostly produces better overall performance.

5.1.7 Replacement Strategy

In light of the evolution process, new offspring produced by the GA operators replace individuals in the population. The average fitness of the population then can be improved over generations. For years, many researchers have proposed and tested various replacement schemes. But basically, they can be seen as two important groups. Individuals in the population are replaced either completely or partially, which are called *generational* and *steady-state* replacement, respectively.

The generational replacement strategy, defined originally by Holland (1975), is to replace the entire population after a new population is generated. Note that such a method allows the possibility that the "best" individual is also replaced, i.e., it does not survive to the next generation. Likewise, individuals that carry important "building blocks" may not survive as well. In contrast, the steady-state replacement scheme aims to replace individuals that are "less fit", usually below average, or only a few individuals at a time (see Beasley and Chu, 1996; Levine, 1996). One advantage of such a strategy is that newly produced offspring can be selected for further reproduction immediately after it is generated. In addition, the "best" solution so far in the population is always kept. Therefore, a GA incorporating a steady-state replacement strategy usually tends to converge faster than the one using the generational replacement strategy.

Based on our experimental experiences, we apply a steady-state replacement scheme in the approach. The population is not replaced by the new offspring as a whole, instead a certain percentage of individuals in the population is replaced by their offspring. In our approach a value around 80% is adopted in the experiments, which is higher than approaches that only replace individuals below average. Limited computational experience shows us that such a method may provide a faster convergence.

5.1.8 Computational Results

Because our approach aims to tackle a real-life complex problem, the efficiency of the algorithm must be justified by computational results. Therefore, several experiments were conducted based on the real-life practical setting of a medium-sized European tourist airline. The input data are the real-life flight schedules of one fleet together with the information regarding the crew members' availability and their home bases and destination airports. An overview of the seven tested instances is shown in Table 4.1, where one can see the scale of the problems by the given figures.

Table 5.1: Disruption scenarios

Category	Scenario	Affected flights	Aggregated affected flights	Affected home bases	Affected crew member	Recovery period (hours)
minor	A1S1	1	1	1	1	15
	A2S2	1	2	1	1	12
	A3S3	2	3	1	2	16
	A4S4	2	2	2	2	20
medium	A5S5	4	5	1	4	24
	A5S6	4	5	1	4	36
	A6S7	4	3	2	4	40
major	A5S8	5	7	1	5	72
	B1S9	13	15	1	13	111

Before elaborating on the details of the observed results, a brief introduction to the testing scenarios is given (see Table 5.1). We classify all disruptions into three main categories by their *scale*, namely, minor, medium, and major disruptions. This can be understood as the total number of affected flights together with those subsequent flights which are influenced by them, indicating how many flights have to be rescheduled at least. In addition, the number of affected home bases and crew members as well as the length of the recovery period are further crucial factors for the difficulty of the instances.

The entire approach has been implemented in ANSI C++, and numerous tests were conducted on a regular PC with Intel Pentium IV 2.2 GHz CPU, 2 GB main memory, running Microsoft Windows XP Professional operating system.

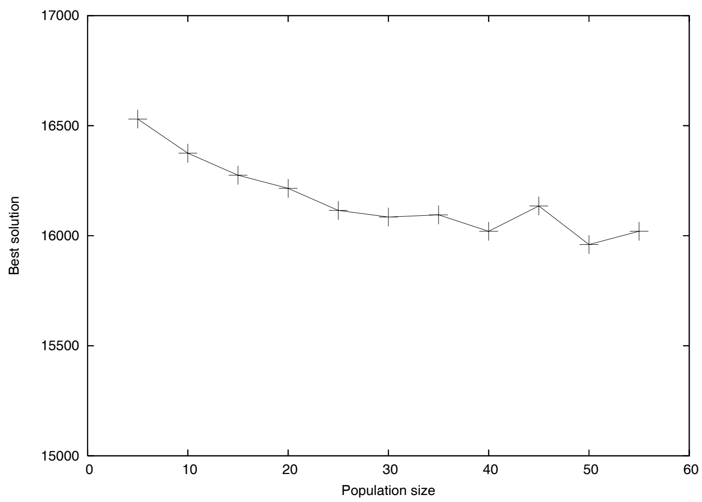


Figure 5.5: Effect of the population size

After a number of tests, the results show that the best parameter setting for one instance does not perform the same in all the other instances, e.g., the number of generations for solving a small instance is relatively lower than that for large instances. Furthermore, a larger population size normally produces better results (see Fig. 5.5, the test is based on instance A5), the best size, however, varies from instance to instance. Based on the results of the experiments, a population size of 25 shows more efficiency for most medium-sized and large-sized instances, while a setting with 45 or more performs better for small instances.

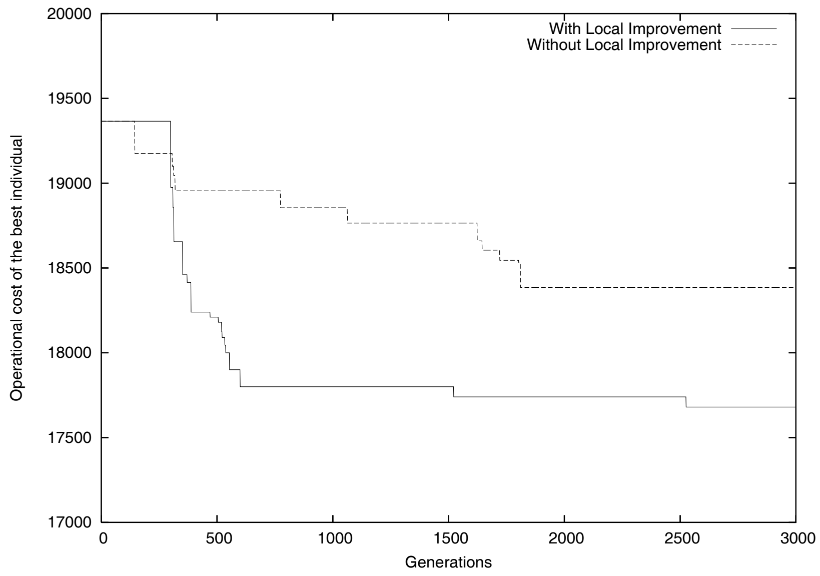
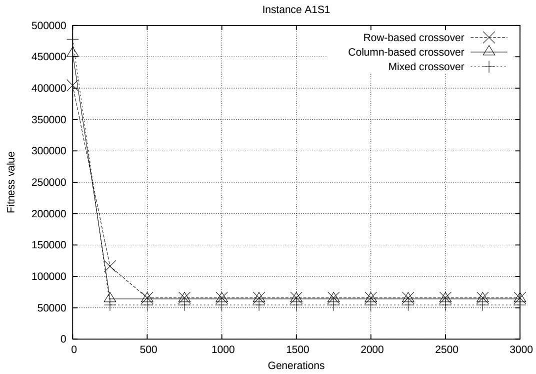


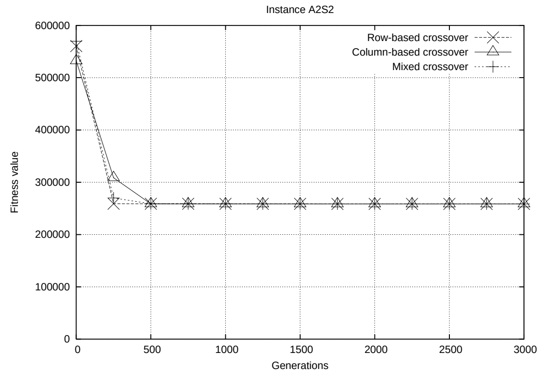
Figure 5.6: Comparison between using and not using local improvement

The local improvement procedure usually produces faster convergence and better results. The comparison indicating the difference between using and not using the local improvement procedure can be seen in Fig. 5.6. In this example, we take the scenario A6S7, and the improvement procedure starts after 300 generations. As one can see, the best individual is always better than the one found by the algorithm without the local improvement after 300 generations. Because the local improvement method maintains feasibility of new offspring, the speed of improving solutions, therefore, can be much faster.

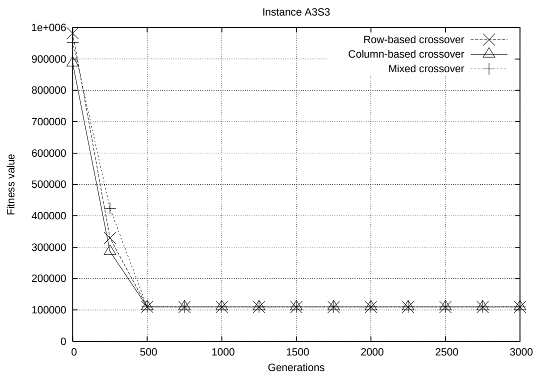
The result produced by the *column-based crossover* operator is slightly better than that produced by the *row-based crossover*. Interestingly, the combination of these two operators with a certain probability (0.5 used in the example) sometimes performs even better than one crossover alone, but it needs mostly more computational time. The best solutions found by using the two different crossovers and the combination of both are listed in Table 5.2. As depicted in Fig. 5.7 and 5.8, the general performance on different instances is presented, in which the algorithms with column-based crossover usually produces faster convergence and better final solutions. Furthermore, the mixture of two crossovers may sometimes



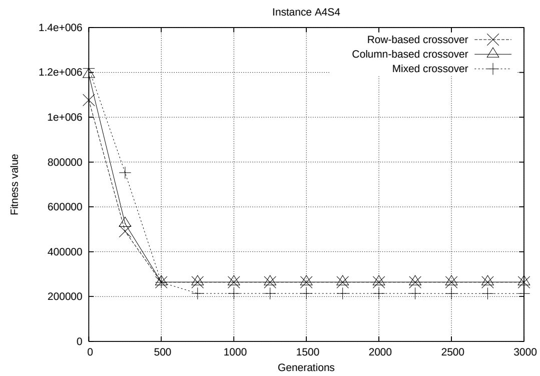
a. Instance A1S1



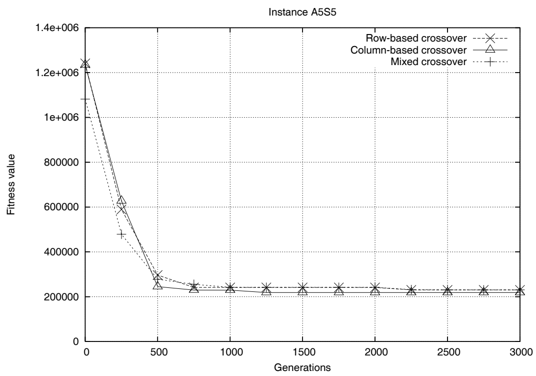
b. Instance A2S2



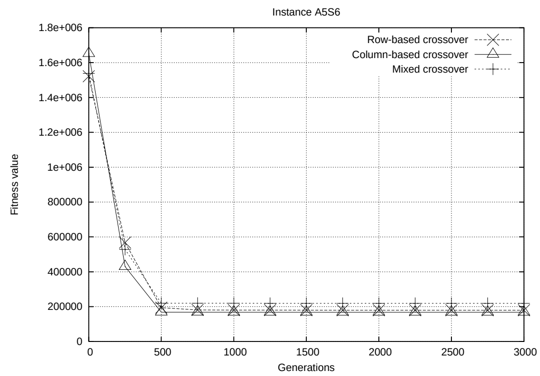
c. Instance A3S3



d. Instance A4S4

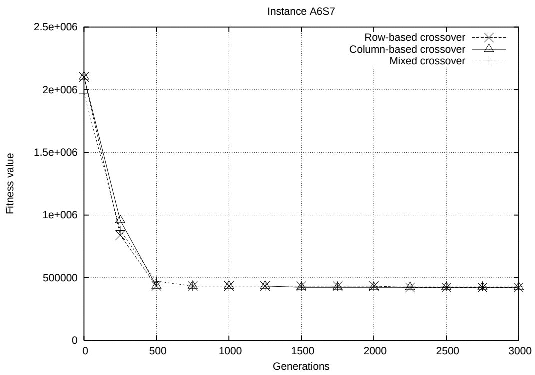


e. Instance A5S5

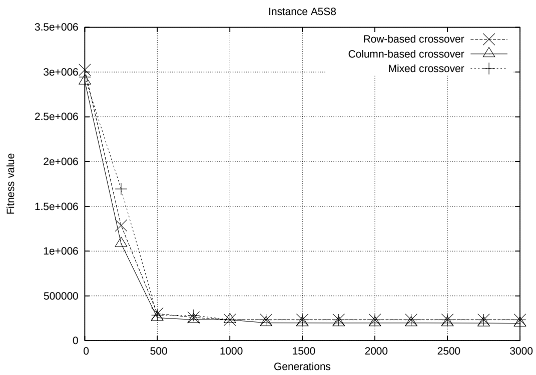


f. Instance A5S6

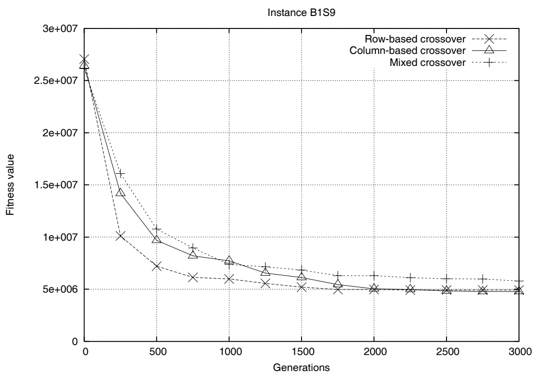
Figure 5.7: The performance of crossover operators on different instances (1)



g. Instance A6S7



h. Instance A5S8



i. Instance B1S9

Figure 5.8: The performance of crossover operators on different instances (2)

Table 5.2: The performance of different crossover operators

Instance & scenario	Row-based crossover		Column-based crossover		Mixed crossover	
	Cost[a]	Time[b]	Cost	Time	Cost	Time
A1S1	1300	9.6	1300	9.6	1300	9.6
A2S2	6950	12.9	6780	13.2	6880	12.8
A3S3	8260	14.2	8260	14.7	8260	14.2
A4S4	11000	14.3	11590	14.5	11320	14.5
A5S5	16390	21.8	16165	22.3	16125	22.3
A5S6	15750	22.4	15890	22.6	15870	22.3
A6S7	18635	22.7	18195	22.7	18430	23.3
A5S8	15965	23.2	16470	23.2	15315	29.9
B1S9	26290	372.8	25945	363.2	26530	482.8

[a]Operational cost, lower is better
[b]Computational time for each generation (millisecond)

find better results than using each one alone.

Regarding the quality of the final solution, the algorithm can find an optimal solution rather fast for small instances with around five hundred generations. But it turns out to be difficult to find an optimal solution for larger problems, e.g., A6 and B1. It is true especially when the recovery period is comparably long, i.e., the total number of involved flight legs is large. However, for large instances the solutions finally found after a certain number of generations are all reasonable and acceptable in terms of the minimization of operational cost and the variation from the original crew schedule. For large problem instances, the algorithm normally needs about two thousand generations, after which the improvement over generations becomes very small and converges slowly. For example, the largest instance B1S9 examined in our approach needs 12.1 minutes to complete two thousand generations. However, other instances normally need about one minute and less generations. The results also show that the final solution, for most instances with minor disruptions, is feasible, and does not produce any further operational cost. In addition, for most cases only a limited number of notifications is required, which is acceptable in practice.

5.2 Constructive Heuristics

In our approach, a greedy algorithm is also developed for solving crew recovery problem in a constructive way. It is inspired by a similar approach applied in the airline crew assignment problem (see Section 4 in Guo et al., 2003), due to the great similarity between the two problems. In this section, we first briefly introduce the algorithm that is used to solve the crew assignment in Section 5.2.1, and then in Section 5.2.2 we give the detail of how this method is applied to solve the airline crew recovery problem.

5.2.1 Multi-weight based Greedy Heuristics for Crew Assignment

In (Guo et al., 2003), a partially integrated approach is addressed to solve airline crew scheduling, including pairing generation and assignment. The so called *crew pairing chain* generation approach already considers crew availabilities, prescheduled activities, and crew requests. The assignment, therefore, may take advantage of such partial integration. The possibility of restructuring pairings can be significantly reduced. Eventually, a dedicated constructive heuristic for the assignment task turns out to be applicable. In this section we first describe the multi-weight based heuristic algorithm (MWHA) for the personalized rostering applied in the airline crew scheduling approach.

After pairing chains are generated, crew capacities have been calculated anonymously and the balancing among crew members is not fulfilled. Thus, a so called *situation-based heuristic* including three phases is carried out sequentially: *initial assignment*, *global balancing* over all home bases, and *local balancing* of each modified home base.

The constructive algorithm is applied within the initial assignment step, whose task is to allocate all given activities for a specific home base among all available individual crew members in terms of best fitting. This is achieved by the decomposition of *patterns* (output of the pairing chain generation) into *atomic pairings* (parts of pairings corresponding to home base to home base trips). Firstly, prescheduled activities are linked to their corresponding crew members. Secondly, several multi-weight based selection strategies are adopted: Each of them aims to map the most "promising" atomic pairing to the most "promising" available crew member within the examined home base. All work rules must be completely satisfied in the context of the individual crew member, e.g., maximum of daily/weekly/monthly flight time, flight duty time, and work time, minimum requirements for rest time between flights and flight duties, and special restrictions on early and late night flights.

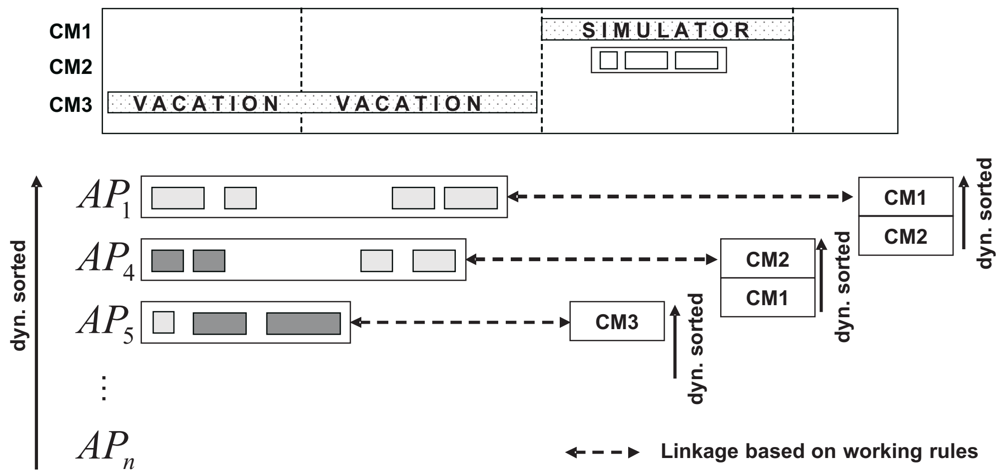


Figure 5.9: Multi-weight based assignment heuristic

Two initiated queues store dynamically all remaining atomic pairings and available crew members for those items, which can be interpreted as the set for linking mutual candidates (see Fig. 5.9). We have adopted two alternative criteria to sort and process the queue of atomic pairings: by start time and by duration. Focusing on start time fills the generated crew schedule from the first to the last day of the planning period (visually from left to right side), whereas sorting based on duration follows the idea of assigning long pairings first because of their low likelihood in finding a suitable free slot later when more pairings have already been assigned.

The available crew member list is dynamically updated since permanent cross-checking has to ensure the consistence of the already assigned rosters of all given crew members at the examined home base. For this queue several weighting strategies are applied as well, such as sorting by decreasing remaining contracted flight hours, or choosing the crew member having less working days left than others for a comparable amount of remaining flight hours.

After selection and assignment of the best mutual candidates, the new situation enforces an update of the queues' composition and resorting by the chosen strategies. Since this simple, but sufficient greedy heuristic is acceptably fast even for larger instances tested, various strategy combinations for queue sorting on both items can be performed in order to choose the best solution.

In the rarely occurring case that few atomic pairings remain unassigned due to the set of granted work rules, a 2-opt procedure with backtracking tries to merge the remaining atomic pairings into the corresponding home base. If successful, no further investigation on those domiciles is required, since these "intra" home base changes do not cause any additional cost in terms of hotel and transit. Otherwise, after handling all home bases as described above, phase two for global rebalancing is required.

5.2.2 Application for Crew Recovery

For the airline crew recovery problem, a similar procedure can be taken to assign affected flights instead of atomic pairings examined in the last section. The basic idea is to repair a disrupted crew schedule rather than to create a completely new schedule. Since it is the recovery process, we may only consider activities within the recovery period RP. The length of DP often is one or several days. Outside such a period, flights and activities are set to frozen activities, which have to remain intact. Furthermore, unlike the crew assignment problem, for such a recovery problem it is not necessary to reschedule all flights within the examined recovery period.

As shown in Algorithm 7, we first remove the original assignment of affected flights from the original crew schedule s. The status of each crew member is checked in terms of his/her current location, elapsed flight hours, working days within the week contractual flight hours, desired destination after the recovery period, and so on. This information is extremely important because we attempt to find crew members who are the most "promising" for the given affected flight.

Algorithm 7 Outline of Multi-weight based Heuristic

Require: $RP \geq DP > 0 \lor F' \neq \varnothing$
Ensure: solution s

```
init(RP)
init(DP)
init(strategy1)
init(strategy2)
init(strategy3)
s ⇐ getOriginalSchedule()
removeAffectedFlight(F', s)
getStatus(W)
Q_CM ⇐ ∅
while F' ≠ ∅ do
   sort(F', strategy1)
   f ⇐ F'[1]
   Q_CM ⇐ findAvailableCM(f, W, s)
   sort(Q_CM, f, strategy2)
   if Q_CM ≠ ∅ then
      w ⇐ Q_CM[1]
      assign(f, w, s)
      F' ⇐ F' − {f}
      updateStatus(w)
      Q_CM ⇐ ∅
   else
      Q_CM ⇐ findPotentialCM(f, W, s)
      if Q_CM ≠ ∅ then
         sort(Q_CM, strategy3)
         w ⇐ Q_CM[1]
         F_conflict ⇐ removeOriginalAssignment(w)
         assign(f, w, s)
         F' ⇐ F' − {f} + F_conflict
      else
         assignToStandbyReserveCM(f, s)
      end if
   end if
end while
output(s)
```

The affected flights set is first sorted in a way that the most "urgent" flights stay on the top. We define the term urgent as how soon the flight will be operated, because the early flights should be considered particularly for a recovery process. Other aspects, such as the duration, may not be as important as the departure time of the flight. For each flight $f \in F'$ in the set of affected flights (also can be seen as unassigned flights), the algorithm tries to find a set of crew members Q_{CM} which is available for serving flight f. Here we say a crew member is available means that he/she is able to operate the flight without violating any rules and regulations. The set Q_{CM} is also sorted using a certain strategy. Possible strategies in this problem setting include the ascending order of incurring operational cost c_{opl}, change cost c_{chg}, working hours h or the combination of all of them. In our approach, we first compare the sum of c_{opl} and c_{chg} and then the working hours h secondly. If there exist available crew members, the best of them is chosen to operate the flight.

In the case that the set Q_{CM} is empty, we need to find another set of crew members in which all crew members may potentially take the flight. Every crew member who stations at the appropriate airport and has assigned flights at that time will be considered as a potential candidate for operating the examined flight. The sorting strategy of such a set is similar to the one used above, but it includes one additional criterion: the number of flights that need to be reassigned in favor of assigning the current one. If the set is not empty, the flight is assigned to the best suitable crew member by producing further unassigned flights($f \in F_{conflict}$). Otherwise, the flight is assigned to standby or reserve crew since there is no way to find an operating crew member who may possibly operate it.

The above process is repeated until there is no affected flight left without being assigned. Based on our computational experience, the algorithm usually require less one minute to find a solution, including solving large problem instances. The generated solution is feasible, but it is also very likely that the solution is not optimal. Despite not the optimal solution, it is applicable in most cases. Furthermore, we found this algorithm is useful in those urgent cases where coordinators may need to save the recovery time by shortening the recovery period, or even disruption period by leaving some affected flights for later rescheduling.

5.3 Comparison of Solution methods

In Chapters 4 and 5, we have presented computational results for the airline disruption management problem with exact optimization, genetic algorithms and constructive heuristics. It is difficult to directly compare their quality, because the problem formulations and the goal functions are not identical. However, some preliminary conclusions will be derived here indicating at least in some typical cases which method might be favorable in a given situation.

The mixed-integer optimization models (4.8 – 4.11) and (4.12 – 4.16) can be solved optimally for minor disruptions with state-of-the-art software, however, the solution takes about one hour (on a state-of-the-art Intel-based PC) [1] for a rather small problem which is too long to wait. Through column generation the solution time can be reduced, and we were able to solve slightly disrupted problems within 10 minutes which is close to being acceptable. For larger disruptions or larger airlines, however, the column generation technique is not able to find an optimal solution within reasonable time either.

The genetic algorithm proposed in Section 5.1 provides acceptable results within a few minutes for small problems, and needs about 10 minutes to compute feasible solutions for the largest test problem. However, this method may be stopped any time, and the best solution so far can be made feasible through a fast correction algorithm. Additionally, with the local improvement procedure feasible solutions can be maintained in the population.

Finally, the constructive heuristic is usually able to compute a feasible solution fast, however, the solution quality cannot be judged directly and may be far below that of the column generation method.

We conclude that it would make sense to include all three approaches in a decision support system for coordinators, and will next work on finding a suitable classification of disrupted situations in Chapter 6.

[1] all computing time refer to a PC with Pentium 4, 2,2 GHz and 2 GB memory

Chapter 6

Disruption Classification and Strategy Mapping

In this chapter, we embark on the task of finding an appropriate strategy that may ease the problem solving in terms of efficiency. As we all know, it is quite natural that knowing more about things that one intends to do may be significantly helpful to achieve goals expected. Such a principle applies for the airline crew recovery problem as well. When disruptions occur, coordinators of an airline have to make concerted efforts to investigate them before making any decisions. They need to know what causes disruptions, how serious disruptions are, what must be essentially achieved, and so on.

Once coordinators learn all the facts regarding the disrupted situation, one of the strategies at hand has to be chosen to deal with the given problem. The decision of selecting a particular strategy may have a dramatic influence on the overall performance of the crew recovery process. However, choosing an appropriate strategy often is a difficult task, since the selecting process is not decisive. In addition, numerous options may be involved, so that a deep understanding of their specific aspects and their impacts are necessary. This drives airlines' coordinators to consider a systematical method that can assist them in making decisions.

Based on our experiments, we concluded in the previous chapters that a decision support system for airline crew recovery should incorporate several solution techniques and help in choosing the right one in each situation. Our first attempt to classify disruptions was based on their severity (minor, medium, major), however, it turned out soon that this rough classification is not helpful in choosing a solution method (see Chapter 2, Section 2.2.4). Thus, in this chapter we present a new classification scheme based on the specific type and, especially, specific goal of a given recovery process. Furthermore, we apply an evaluation and selection procedure able to incorporate several goals with selected priorities.

This chapter begins with a description of the disruption classification in Section 6.1, in which disruptions are examined so as to build relationships with criteria of given strategies. In Section 6.2, we propose a strategy mapping approach using Analytic Hierarchy Process (AHP), by which a particular strategy is adopted out of a bundle of alternatives to handle the given disrupted situation. In the end, we give an example of the overall mapping process in Section 6.3.

6.1 Disruption Classification

As already described in Chapter 2, there are a wide range of possible causes which lead onto the development of disruptions. Different sources may have divergent impacts on the operating of flight schedule. It is important to learn what actually causes disruptions when they occur, but more indicative information lies in what airlines intend to attain after the recovery process under the given circumstance. For instance, in a particular situation where an unserious disruption causes small changes in the crew schedule, an airline may prefer to find a recovery solution which does not introduce additional operational cost and only few changes. In contrast, in other serious disruptions the airline has to face the tough challenge and emphasize a rapid recovery process. Despite the fact that a fast recovery process may produce the best solution in terms of cost, the airline can save time in order to bring back the normal operation as practical as possible.

There are several factors that influence the decision of choosing a proper strategy. Regardless the actual source which causes disruptions, the following measurement may give ideas how serious a given situation is:

- M_d, the number of delayed flights
- M_c, the number of cancelled flights

- M_n, the number of new flights that need to be operated
- M_r, the number of crew members whose schedule is disrupted
- M_m, the number of available crew members
- M_a, the number of daily flights in average
- M_u, the length of the disrupted period

Therefore, we give a measurement vector M for each disrupted situation, which can be presented as

$$M = (M_d, M_c, M_n, M_r, M_m, M_a, M_u)^T \tag{6.1}$$

Given two disruptions A and B and their associated measurement vectors M_A and M_A, comparing the severity of the two disruptions becomes the task of evaluating the values in the measurement vectors. Therefore, we give a severity vector S which includes two elements: s_f and s_w. s_f denotes the percentages of daily flights that are affected by the disruption, while s_w means what percent of crews are affected by the given disruption. We give the calculations of them as

$$S = \begin{pmatrix} s_f \\ s_w \end{pmatrix} = \begin{pmatrix} \frac{M_d+M_c+M_n}{M_u \, M_a} \\ \\ \frac{M_r}{M_m} \end{pmatrix} \tag{6.2}$$

therefore, two severity vectors, S_A and S_B, are computed. Disruption A is severer than B, if $s_{fA} > s_{fB}$ or $s_{fA} = s_{fB} \wedge s_{wA} > s_{wB}$.

As already discussed in Chapter 2, we define three groups of disruption severity: minor, medium and major. To distinguish disruptions in this way, we give two bounds $S1$ and $S2$, which are the bound between minor and medium and the bound between medium and major, respectively. We define a disruption is minor if the values of s_f and s_w are both lower than $S1$ (e.g., $S1 = 0.05$). A disruption is considered as major, if one or both have a value larger than $S2$ (e.g., $S2 = 0.15$). Others are thus treated as medium disruptions. Such a calculation may differ from airline to airline, since the scale and the operation of every airline differ significantly. Accordingly, the difference may be reflected by changing the two bounds $S1$ and $S2$ fitting to an airline's individual scenario.

Basically, the calculation of a disruption's severity may give coordinators the impression how serious it is. A series of considerations or actions may be conducted based on such information. However, after a variety of experiments, such a classification proposal turns out to be not intuitive for a coordinator to choose an appropriate solution method and its corresponding setting. Since there is no clear and direct connection between solution methods and the severity of disruptions, it is ambiguous to select one single solution method for a given disruption. As a solution method, together with its specific setting, may have higher potential to achieve a particular goal over others, we propose another classification scheme based on the specific type and goal of a given recovery process. In other words, we examine the type of the recovery process and give a set of goals associated with their importance. The Analytic Hierarchy Process (AHP) is then applied to select a method and its specific setting out of a number of combinations, which may potentially achieve the goals examined previously. In the following sections, we will give so-called strategy mapping procedure to single out an appropriate strategy for a given disruption, in which the AHP technique is utilized.

6.2 Strategy Mapping

In this section, we will first give the foundation of the AHP in Section 6.2.1 and then describe the steps of applying the AHP in the setting of the airline crew recovery problem.

6.2.1 Basis of the Analytic Hierarchy Process

Many decision making processes involve preferential selection among a finite set of alternatives or courses of action. To handle such a situation, the Analytic Hierarchy Process was developed by Saaty (1980) to provide the prioritization of alternatives through evaluation of a set of criteria elements. It helps structure decision making processes in complex environments by using ratio scales to quantify subjective judgments. For years it has been used in a wide variety of applications and has proven to be an accepted methodology in many areas.

The AHP can be basically described as the following steps:

Step 1 The given problem is decomposed and structured as a hierarchy of a goal, criteria, sub-criteria and alternatives. Such a hierarchy represents the relationship between elements of one level with those of another level below. Saaty suggests to build a hierarchy by working down from the goal as far as one can, and then working up from the alternatives until the levels of two processes are linked. With such a hierarchy, we can start to establish a pairwise comparison matrix.

Step 2 Decision makers or experts give pairwise comparisons of alternatives on a qualitative scale according to the given hierarchy. A comparison can be rated as 1 (equal importance), 3 (slightly strong importance), 5 (strong importance), 7 (very strong importance), 9 (absolute importance), values in between (2,4,6,8, fuzzy intermediate values) and reciprocals of above (dominance of second alternative). Below shows how to establish the pairwise comparison matrix.

Let $C_1, C_2, ..., C_n$ be the set of elements in a hierarchy. $w_1, w_2, ..., w_n$ are their weights of influence. a_{ij} indicates the strength of C_i when compared with C_j. Therefore, a reciprocal matrix A can be built as follows

$$A = [a_{ij}] = \begin{bmatrix} 1 & a_{12} & \cdots & a_{1n} \\ 1/a_{12} & 1 & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 1/a_{1n} & 1/a_{2n} & \cdots & 1 \end{bmatrix} \tag{6.3}$$

where each element $a_{ij} = w_i/w_j$ $(i, j = 1, ..., n)$.

Step 3 Once the pairwise comparison matrix A has been constructed, the eigenvalue method is used to rank the elements. The principal eigenvalue and the corresponding normalized right eigenvector $\hat{c}$ of the matrix A gives the priority of the criteria.

Step 4 Since the value of a_{ij} usually is an estimate as a judgment, the consistency of the matrix A must be tested. The *consistency index (CI)* and *consistency ratio (CR)* are proposed to check the consistency, which are calculated as

$$CI = (\lambda_{max} - n)/(n - 1) \tag{6.4}$$

$$CR = CI/RI \tag{6.5}$$

where λ_{max} is the maximum eigenvalue of the matrix A and *random index RI* is randomly generated reciprocal matrix from the scale 1 to 9.

Step 5 Firstly, the ratings of each alternative are multiplied by the weights of the sub-criteria, which are aggregated to get local ratings. Secondly, the local ratings are multiplied by the weights of the criteria, which are then aggregated to obtain the global ratings, namely priorities of the alternatives.

6.2.2 Criteria in the Airline Crew Recovery Problem

In the context of the airline crew recovery, in order to properly recover crew schedules from disruptions, airlines need to choose a proper strategy out of a set of alternatives with respect to the given disrupted situation. Usually, crew managers or coordinators of an airline make decisions based on their rule of thumb and practical experiences. In such a process, the AHP certainly shows its potential to assist this decision making. Therefore, we propose an AHP based strategy mapping method to assist coordinators selecting strategies.

The main criteria in this problem are usually concerned by airlines with respect to many aspects, such as economic perspective, efficiency and convenience. In our approach, we give the following four criteria:

Additional cost The extra money that needs to be paid for repairing the disrupted crew schedule. The calculation of such a cost can be found in (4.1);

Solution time The computational time to find the final solution. In other words, how fast does an airline recover its crew schedules;

Notifications The number of crew members that need to be notified because of the schedule updates. It also can be seen as changes examined in previous chapters;

Updated period The duration of the period starting from the time of the first updated flight assignment to the last one;

Disturbance to crew The number of schedule changes to crew members and unbalanced workload among them.

6.2.3 Solution Strategies

The strategies are usually defined by experts from both airlines and the development organization. A finite strategy set S is built based on the empirical experience after a large amount of experiments. A strategy can be understood as a combination of a disruption scenario setting and a solution method together with its parameters. It reflects the given disruption scenario and the result expected.

After a disruption is identified, the corresponding disruption scenario has to be defined. It normally includes the following aspects:

- The length of the recovery period RP. It can be set to a relatively long period as the disruption does not seriously affect the operation and the airline has enough time to consider a fairly good or optimal recovery solution. In contrast, it may be set to a very short period, e.g., within the day, because of considerable time pressure and the large number of flights and crew members involved.
- The number of home bases involved. This can sometimes be smaller than the total number of the home bases of the airline, since it is practical if the problem can be localized. Therefore, in this case it is unnecessary to involve all home bases into the recovery process, which can reduce the problem size significantly.
- The number of the crew members. Similar to the previous one, the number of crews involved in the process can be reduced to a certain degree.

Basically, two groups of solution methods are considered, exact optimization methods and heuristic based methods. Further explanations are given as follows:

Exact Optimization based Strategies As introduced in Chapter 4, this group of solution methods includes the direct solving approach and a column generation based method. Generally, regarding the size of the problem two criteria may have a great impact on the overall performance: the length of the recovery period (the period within which the schedule needs to be recovered), and the number of home bases and crew members involved in the process.

Table 6.1: Example strategies for solving the airline crew recovery problem

Strategies:	CGLO	CGLA	CGSA	GALO-SS	GALA-SS-LI
Key aspects:	.Col-Gen .LRP[a] .OHBs[c]	.Col-Gen .LRP .AHBs[d]	.Col-Gen .SRP[b] .AHBs	.GA .LRP .OHBs .Solution Seeding[e]	.GA .LRP .AHBs .Solution Seeding .Local Improvement

[a]Long recovery period
[b]Short recovery period
[c]Only those home bases affected by disruptions
[d]All home bases in the instance
[e]Seed some individuals representing original schedule with little changes

Heuristic based Strategies The group of heuristic based methods mainly include two heuristics introduced in Chapter 5. One is a constructive heuristic and the other is a hybrid genetic algorithm incorporating a number of sub-processes with knowledge of this particular problem. The performance of the algorithms is subject to numerous parameters proposed in our approaches. Taking the GA based method as an example, the chosen variation operators and their corresponding rate have dramatic impacts on the overall problem solving.

As mentioned previously, the finally selected strategy is the combination of the disruption scenario, a solution method and its relevant parameters. They can be enumerated by combining them together. Certainly, there are a number of possible combinations. But only a subset of them is typically considered in practice. Table 6.1 gives some example strategies, e.g., the strategy CGLO denotes the use of the column generation method, and it only recovers the disrupted schedules of home bases that are affected by disruption, with a 4 or 5 days long recovery period.

6.2.4 Strategy Mapping Process

Once we have defined criteria and have built all possible strategies that deal with disruptions, we are able to establish the hierarchy. A sample hierarchy can be seen in Fig. 6.1), which has three levels, 5 criteria and n available strategies. With the influence weights and the comparisons in between criteria, we can find a best-fitted strategy by going through the five steps introduced in Section 6.2.1. Below we give the detail of applying the AHP technique to map a strategy onto the given disrupted situation.

After investigating occurred disruptions, the pairwise comparisons are conducted between each two criteria with respect to their importance or influence to the final decision. A criteria matrix $A = [a_{ij}]$ is built with the weight ratios a_{ij}. The normalized principal right eigenvector $\hat{C}$ of A represents the priority of those criteria of a proper recovery. We compare the strategies on each of the criteria by examining how efficient one strategy may handle the given problem. Same to the number of available strategies, n strategy matrices $W_i, i = 1, 2, ..., n$ are produced, and the priority vector $\hat{W}_i, i = 1, 2, ..., n$ for each strategy can be calculated in the same way described above. The final priority vector $\hat{W} = [\hat{w}_i,\ i = 1, 2, ..., n]$ can be thus calculated as

$$\hat{W} = [\hat{W}_1\ \hat{W}_2\ \hat{W}_3 ... \hat{W}_n]\hat{C} \tag{6.6}$$

where $\hat{W}$ gives the final ranking of the strategies with respect to the given disruption. The highest priority value denotes the proper strategy for the examined situation. It can, therefore, be chosen as the most suitable strategy and method to solving the current disruption.

6.3 Case Study

A case study is presented to demonstrate the strategy mapping and proposed solution method. The instance is from a European airline with multiple home bases inside Germany and more than 30 destinations spreading around Europe. In this case study, we consider three criteria: additional cost, solution time and notifications. Only two strategies are used for the purpose of simplicity, e.g.,

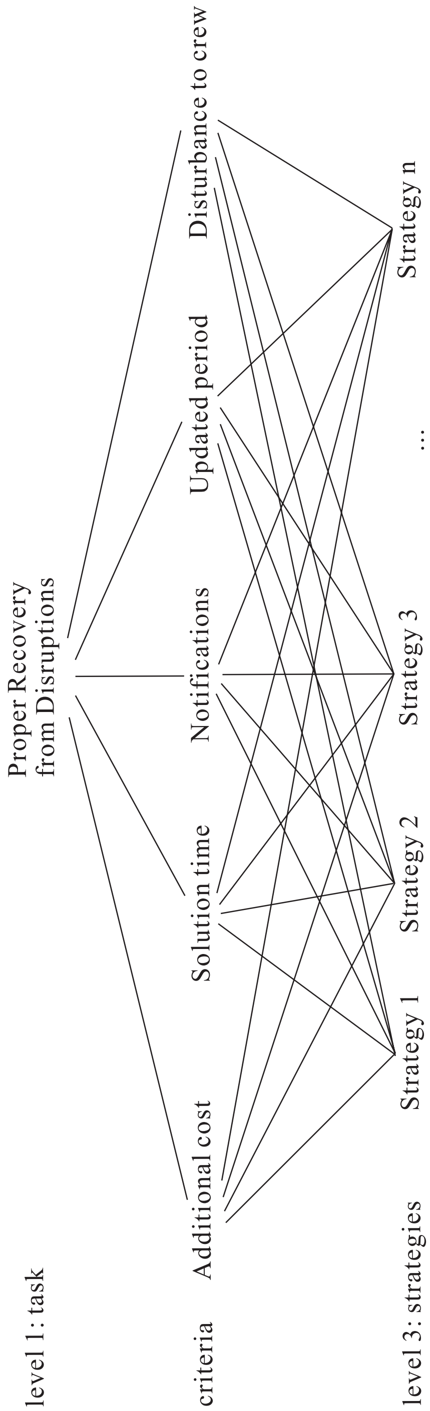


Figure 6.1: A sample AHP hierarchy for strategy mapping, with 5 criteria and n strategies

Table 6.2: Pairwise comparison matrix of criteria

	additional cost(AC)	solution time (ST)	notifications (N)	priority
AC	1	1/3	5	0.38
ST	3	1	5	0.54
N	1/5	1/5	1	0.08

Table 6.3: Pairwise comparison matrix of strategies on the criteron AC

AC	CGLA	GALO-SS-LI	priority
CGLA	1	2	0.667
GALO-SS-LI	1/2	1	0.333

CGLA and GALO-SS-LI. Furthermore, a disruption situation is given with the measurement vector $(2, 1, 2, 2, 188, 85, 6)^T$, which indicates a minor disruption on a comparably large size of problem instance.

A hierarchy can be built as Fig. 6.1 by eliminating two criteria and replacing strategies with only two example strategies: CGLA and GALO-SS-LI. The pairwise comparison matrix of criteria can be seen in Table 6.2, whose values are set by coordinators of the airline based on their empirical experiences. The normalized priority vector, therefore, is $\hat{C} = [0.297, 0.617, 0.086]$.

The comparisons between strategies are done based on criteria, with one matrix being created for each criterion (see Table 6.3, 6.4 and 6.5). With the above method (6.6), we obtain the final priority vector $\hat{W} = [0.371, 0.626]$ which clearly shows that the second strategy GALO-SS-LI is superior to CGLA by reflecting the priorities among the introduced criteria. After a variety of experiments were

Table 6.4: Pairwise comparison matrix of strategies on the criteron ST

ST	CGLA	GALO-SS-LI	priority
CGLA	1	1/3	0.25
GALO-SS-LI	3	1	0.75

Table 6.5: Pairwise comparison matrix of strategies on the criteron N

N	CGLA	GALO-SS-LI	priority
CGLA	1	1/3	0.25
GALO-SS-LI	3	1	0.75

conducted, the results show that the strategy GALO-SS-LI produced a slightly worse solution than that the CGLA could find. But the solution time is dramatically reduced into the acceptable period of time.

6.4 Summary

Because of diverse disruptions, in this chapter we propose a disruption classification process and a strategy mapping procedure using AHP to cope with the difficulty arising from specific problem scenarios. The disruption classification can help coordinators understand disruptions better and form more seasonable and specific strategies. The strategy mapping method turns out to be very effective to individual problem instances comparing with approaches with the generic setting.

Chapter 7

A Decision Support System for the Airline Crew Recovery

On top of having strengths, human decision makers certainly have weaknesses to make crucial decisions within a complex business environment. As documented in literature, humans frequently rely on simplifying heuristics rather than normative methods that are able to solve problems optimally, due to limited information processing capabilities. The result from a manual decision process, therefore, may have certain drawbacks which are primarily led by human biases and the lack of information. However, the emergence of decision support systems changes the way humans make decisions, since a DSS is able to provide significant support to decision makers' information processing.

For the airline crew scheduling and recovery problems, a DSS can fundamentally assist airline planners and coordinators. For both problems, they have to put a lot of effort into the generation or the update of crew schedules, since a huge amount of data and the complex problem structure make the process extremely difficult. In the planning phase, for example, it is a great mental challenge for humans to manually find an optimal crew schedule out of thousands of flights and hundreds of crews. Likewise, it is quite difficult for coordinators to find an alternative crew schedule in a short period of time, which has limited additional costs and does not produce too much disturbances to the future operation and crew members. Therefore, a DSS needs to be deployed to assist them and propose applicable solutions.

In this chapter, we begin with a general description of DSS and reason the necessity of applying a DSS for the examined problem in Section 7.1. In Section 7.2 we give the requirements of developing a DSS for both planning and recovery problems. It is then followed by a dedicated DSS architecture in Section 7.3 and the reviewing of the process in which airlines handle disruptions using the DSS. Finally, we give a brief summary concerning the proposed DSS in Section 7.5.

7.1 What Can a DSS bring to the Airline Crew Management?

With wide range of applications, decision support systems have played pivotal roles within a variety of decision making processes, since its early introduction in 1970s. The concept of DSS keep developing over the years, especially after the confluence of different concepts, such as OR/MS, Information Systems, Computer Science. Generally speaking, a commonly accepted explanation of DSS is that it is the combination of Computer-based Information Systems and OR/MS. However, precisely defining DSS is not a straightforward task, although it was done by many researchers all over the world. But there is no single definition of DSS that is agreed by everyone.

A DSS can be understood in many ways. For instance, it may be defined in terms of problem type, system function, interface characteristics, usage pattern, system components, development process etc. However, it is approved by researchers that two aspects must be given to a DSS: (1) it assists human decision makers to make decisions; and (2) it does not make decisions or replace humans completely. In this thesis we adopt a broad definition that was proposed in Silver (1991):

> A *Decision Support System (DSS)* is a computer-based information system that affects or is intended to affect how people make decisions.

The framework of DSS addresses the numerous elements involved in providing computer-based decision support. It provides and combines underlying technologies, decision making processes, system architectures, designs, analysis, visualizations, evaluation, implementation etc. Each of them identifies itself in its own way, and meets decisional needs differently. Instead of simply listing all functional capabilities, describing a DSS requires considering how the individual capabilities fit together to form a whole, contemplating the likely effects on decision makers' behavior.

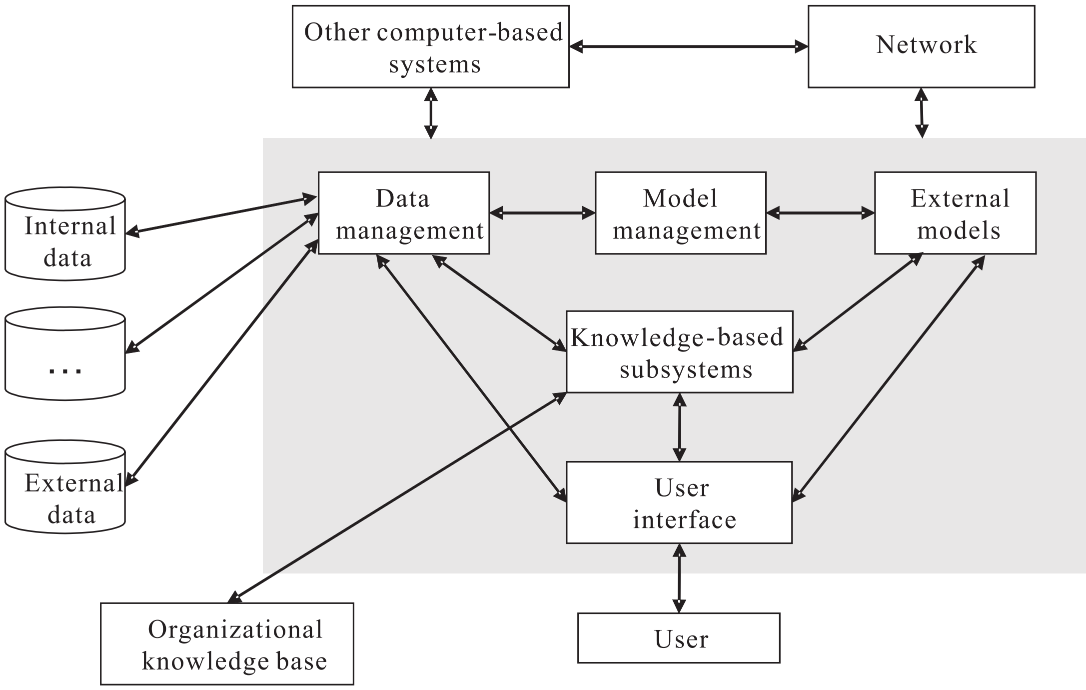


Figure 7.1: The general structure of a DSS

Basically, a DSS consists of many components or subsystems which provide distinctive functionalities. Turban and Aroson (2000) identify four elementary components of a DSS: *data management*, *model management*, *knowledge-based management* and *user interface*. As Fig. 7.1 depicts, the general structure of a DSS proposed in the book shows interrelationships between these four components and how they are connected with other computer-based systems through various types of networks.

As mentioned in Chapter 2, planners and coordinators in airlines make concerted efforts to generate crew schedules that utilize one of the most cost intensive resources — flight crew. Researchers are bestirred to develop a large number of computer-based systems that assist airlines in managing resources more efficiently and systematically. In terms of quantity, the literature concerning decision support systems for the airline crew (re)scheduling problem has no shortage of mathematical models, algorithms, and the like. But there is a lack of the full description of a DSS that may take full advantage of DSS and its substantive elements.

In this chapter, we present a decision support system that supports both the airline crew scheduling and the crew recovery process during irregular operations for a European tourist airline. Apart from generating crew schedules ahead of their actual operations, the system is also designed for recovering disrupted crew schedules, which produces an updated crew schedule at minimum additionally required cost with respect to various restrictions. The system takes advantage of mathematical models and various solution methods proposed previously, and measures disrupted scenarios for the purpose of the efficient problem solving.

7.2 Requirements for Decision Support Systems

It is well acknowledged that various types of decision support systems were deployed in airlines to schedule their flights and crews. However, only few airlines have set up a complete decision support system for managing the entire process of the crew scheduling, especially the problem of rescheduling their crews in disrupted situations. One example development is described by Yu et al. (2003), in which they presented a commercial product, CrewSolver, applied successfully in Continental Airlines in the U.S. to generate globally optimal, or near optimal crew recovery solutions. Nevertheless, manual recovery procedures can be often observed in most airlines all over the world, particularly in medium-sized, small-sized tourist airlines and low-cost airlines.

Therefore, there is an apparent lack of such a DSS that can systematically handle frequent disruptions to crew schedules. The desired decision support system should be able to present the examining problem and propose recovery solutions with the intervention of humans. The duration of such a process should be as limited as practical, which may potentially save costs and prevent further chaos.

The cost of recovery actions must be minimized in terms of their imposed operational costs and the influence on crews and future operations. Furthermore, the system has to interact with the crew schedule planning process, since there exist a great deal of shared data and identical processes. Therefore, the two systems should work as a whole, or function as a complete system that covers the entire life circle of a crew schedule.

Clearly, due to the different problem structures the sub-systems for the two problems differ from each other in terms of their functionalities required. Table 7.1 shows a detailed comparison concerning their functionalities between the crew scheduling in the planning phase and the crew recovery in operations time. We discuss the differences within seven groups of components: graphical user interface, data communication, solution methods, simulation, visualization, publication/notification and statistics.

As described in Chapter 2, Section 2.1, the airline crew scheduling process determines the assignment of flights to crew members during the planning phase. The input of the process includes the flight schedule, aircraft rotations, individual crews' pre-scheduled activities, cost structures, contract data, rules set, and so on. The output, however, is the final crew schedule which usually covers one or half a month long period and requires minimum operational costs.

Therefore, the DSS has to process the complex data with different structures. It needs to involve dedicated solution methods in combination with appropriate strategies. Because of the huge amount of data and the cost intensive aspect, the DSS is required to include an optimal solution method which produces economically optimized crew schedules. The generated solutions still require intensive testing and evaluations in terms of their quality and robustness. Once the resulting schedule is evaluated by conducting simulations, it is officially published to crews and visualized in a clear and understandable way. Furthermore, statistics help to uncover latent problems by analyzing history schedules and their operations.

For the crew recovery problem, the input differs from the planning process. In addition to the flight schedule, history crew schedules and static data, the CRP has to consider the given disruptions, updated flight schedules and the originally planned crew schedule. The output is a repaired crew schedule which is applicable

Table 7.1: DSS functionality for the airline crew scheduling and recovery

DSS Component	DSS Functionality	
	Crew Scheduling (planning phase)	Crew Recovery (operational phase)
Graphical User Interface	. Generation of crew schedules by the planning department	. Monitoring and handling of disruptions by OCC . Providing multiple strategies and their configuration
Data Communication	. Flight schedule . previous crew schedule . Static data regarding crew, airports, aircraft etc	. Detected disruptions . updated flight schedule . Original crew schedule and previous schedule . Static data regarding crew status, airport, aircraft etc
Solution Methods	. Dedicated methods with specific configuration . Optimality is required	. Dedicated methods . Preprocessing . Real-time solution methods are more important than optimality
Simulation	. Evaluation of crew schedules regarding robustness and bottlenecks	. Analysis of impacts regarding current and further possible disruptions in the future
Visualization	. Representation of resulting crew schedules in different steps	. Representation of recovered crew schedules . Visualization of variations from original schedules . Visualization of conflicts in the schedule
Publication/ Notification	. Regular publication of crew schedules	. Notifications to affected crews . Consideration of crews' locations
Statistics	. Generating schedule quality indicators	. Generating recovery quality indicators . Review on recovery strategies, impact analysis, detection of disruption regularities etc

in the given disrupted situation and does not impose too much further costs and disturbances. Furthermore, the recovery process needs to be completed within a reasonable length of period, which introduces further difficulty comparing with the CSP.

Accordingly, the DSS for crew management in the operational phase has to deal with not only the originally scheduled activities and crews, but also unexpected events which disturb the normal operation of the crew schedule. Solution methods that can find solutions within short period of time should be accorded high priority. It is also important to reveal disruptions' impact on the remaining schedule and to act quickly and appropriately towards the full schedule recovery. After the recovery process, the resulting crew schedule is visualized to represent the new schedule together with its changes and possible conflicts. Moreover, the subsequent analysis has to be carried out to review the overall performance of the recovery process and find out the efficiency of individual strategies. The detection of possible disruption regularity may help airlines to generate more robust crew schedules in advance and to handle disruptions proactively.

Because of the complexity of the crew recovery process, the DSS ought to classify disruptions and investigate possible strategy which may find recovered solutions more effectively. Due to the large number of possible solving techniques and strategies, the system needs to manage them in a way that users can select one of them with respect to the given situation. Within each strategy, a solution technique is prioritized and customized in order to handle a specific type of disruption more efficiently and effectively.

To sum up, such a decision support system for the airline crew recovery problem should be characterized by the following: (1) It assists coordinators to classify disruptions and adopt a suitable strategy to handle them; (2) It proposes recovery solutions to coordinators with respect to all the requirements; (3) It analyzes the performance of the recovery and provides possible further improvement.

7.3 A Dedicated Decision Support System Architecture

In this section, we first propose a decision support system architecture that is designed to cover the complete airline crew management process, including the planning and the operational phase. This makes it possible to handle the two different tasks within one infrastructure in which several common components can be shared for both problems. It thus may provide higher levels of interaction between each stage of the entire process, and general problem solving techniques can be incorporated seamlessly into the system. Consequently, in addition to the benefit from the application of sophisticated solution methods, better results can be achieved in such a system level.

Fig. 7.2 shows the architecture of the decision support system for the airline crew management. It emphasizes the two major stages in the process of the airline crew management, namely crew scheduling in the planning phase and crew recovery in the operational phase.

The DSS architecture presented above can be seen as three tiers from the left side to the right side: users, core components and data. In the following subsections, the three parts will be described in detail.

7.3.1 Users and User Interface

In the context of the airline crew management, two main groups of users are involved: planners and coordinators. We differentiate them by their divergent work content in different stages. Planners mainly work in a planning department where they create crew schedules, while coordinators solely work in operations control center where they carry out the reparation of disrupted crew schedules.

The graphical interface (GUI) is the connection between users and the system, which allows them to interact with the different components of the DSS. Both groups of users use similar GUI as the two problems have great similarity except that few distinctive functions are required by each problem (see Table 7.1).

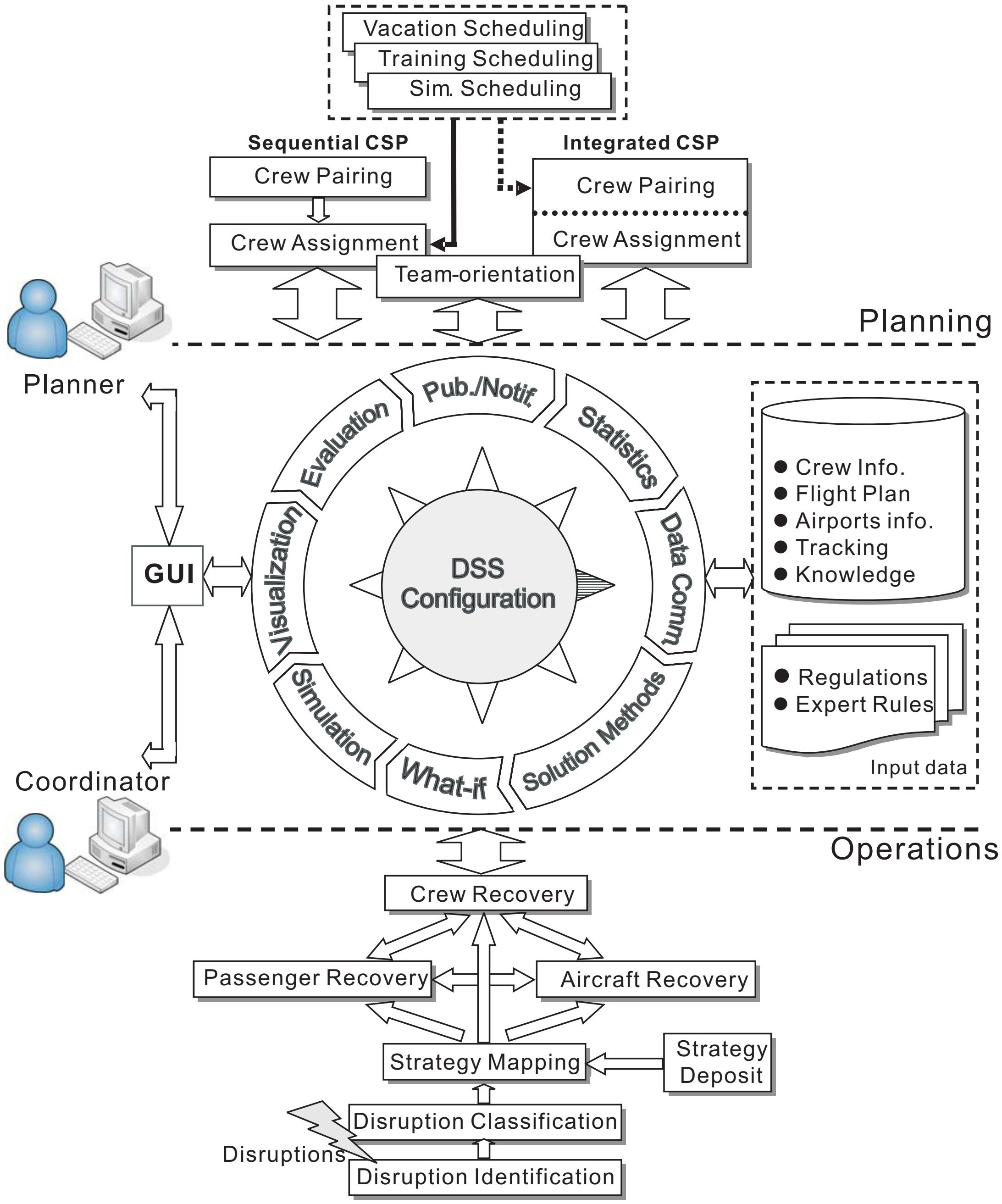


Figure 7.2: The system architecture of the DSS

7.3.2 Core Components

The core components characterize the DSS and provide the individual functionalities. In our approach, we take nine important components into account, and incorporate them into the system. In the following, we briefly discuss these essential components.

DSS configuration: The system configuration is a central component as it constitutes the basis of the DSS. A bundle of functions, parameters and settings are set up with respect to the given situation. It reflects the definition of the airline's objectives and scheduling process. Based on a specific configuration, other components provide their intended functionalities.

Data communication: This component provides the transferring of data between the components, and also with other systems or databases. Because of the high similarity of input data for both of the crew management tasks, a fully shared component can be built for both of them.

Solution methods: Based on the objectives of the task and the given parameters, a set of solution methods is provided. This includes mathematical programming, network flow model, column generation, constructive algorithms, local search and population based heuristics, e.g., genetic algorithms, simulated annealing etc. Each of them can be customized into a specific setting and therefore may solve problems more efficiently.

What-if analysis: This component examines alternatives or individual changes of decisions by presenting possible consequences. With such a component, planners or coordinators may test different problem settings, which introduce alternative solutions.

Simulation: Since the operation of an airline is not deterministic, the application of a simulation component is very useful. With simulation, it is possible to evaluate more stochastic characteristics of crew schedules, e.g., robustness against delays, and their impacts on the entire system. For crew recovery, it can test how well the recovered crew schedule will be operated.

Visualization: Both crew scheduling and rescheduling problem include a huge amount of data with complex structures. A proper visualization helps the user to understand the underlying information more easily. The essential goal of the visualization in our approach is to present crew schedules in a way that adequate information is included and they can be easily understood. In addition, changes in a recovered crew schedule are also shown indicating the new assignment.

Evaluation: This component is responsible for the complex evaluation of alternative solutions, through which it supports the final decision making. After several solutions are generated, an evaluation scheme has been defined which determines the pros and cons among the alternatives and provides a concrete suggestion on how to react in the current situation.

Publication/Notification: All decisions that are made during the crew management need to be published. It has to be done in a way that all individual crew members involved in the scheduling process or affected by the updated schedules are informed accordingly by printouts or via terminal stations at their current location.

Statistics: This component provides the analysis of history data which may produce additional benefits. Derived from the experienced problems, expert rules can be extracted and future disruptions might be avoided before their occurrence. For the recovery problem, it investigates the performance of each recovery, which may help airlines build more reasonable recovery strategies and methods.

7.3.3 Data

On the right side of the architecture (Fig. 7.2), we present several types of data involved in the crew management process. They are basically divided into two groups: static data and rules and expertise.

Static data: For the crew management the input data is stored in several databases: They include information of crew members (e.g., individually contracted flying hours, vacations, pre-scheduled activities and home bases), the flight plan (e.g., flights with arrival and departure times, requested crew qualifications and fleet requirements etc.), and airport information (such as landing capacities, hotel availability). Furthermore, a tracking database provides the real-time data as it is executed during operations which also monitors possible disruptions such as delays, cancellations, crew sickness/absentness etc. A knowledge base covers all less structured information, such as general guidelines and concepts, specific knowledge, and so on.

Rules and expertise: In addition, regulations and rules imposed by governments, union agreements or company internal rules are collected and have to be satisfied during the crew scheduling and rescheduling. Furthermore, expert rules are usually created taking specific situations into account. They are usually defined based on the users' experience.

7.4 Crew Recovery Process Flow

As the focus of this work is to solve the airline crew recovery problem, in this section we will describe the use of such a decision support system from a coordinator's perspective. Especially, we will show the process flow of how they handle disruptions.

As shown in the process flow depicted in Fig. 7.3, a preprocessing step is taken to investigate disruptions and possible strategies. Disruptions are classified and the set of goals of the recovery is prioritized with respect to their characteristics. Usually, the number of strategies at hand is increasing gradually through examining more and more disruptions. The more disruptions an airline experienced in the past, the more possible and suitable strategies can be defined by combining different solution methods and the parameters' setting.

Such a preprocessing may, therefore, reduce intensive computations by identifying a proper strategy based on the evaluation of given disruptions. The chosen strategy customized with the eligible method and its dedicated parameters thus may cut down the solution time of solving the problem. The detailed introduction of the disruption classification and the strategy definition can be found in Chapter 6.

The preprocessing procedure can also be seen as the process that initializes the configuration of the DSS. For the airline crew recovery problem, the DSS configuration involves mainly three parts: airline scenario, instance, and strategy (see Fig. 7.4). An airline scenario includes a set of regulations and rules, together with the airline's policies. It can be seen as the basis of the problem

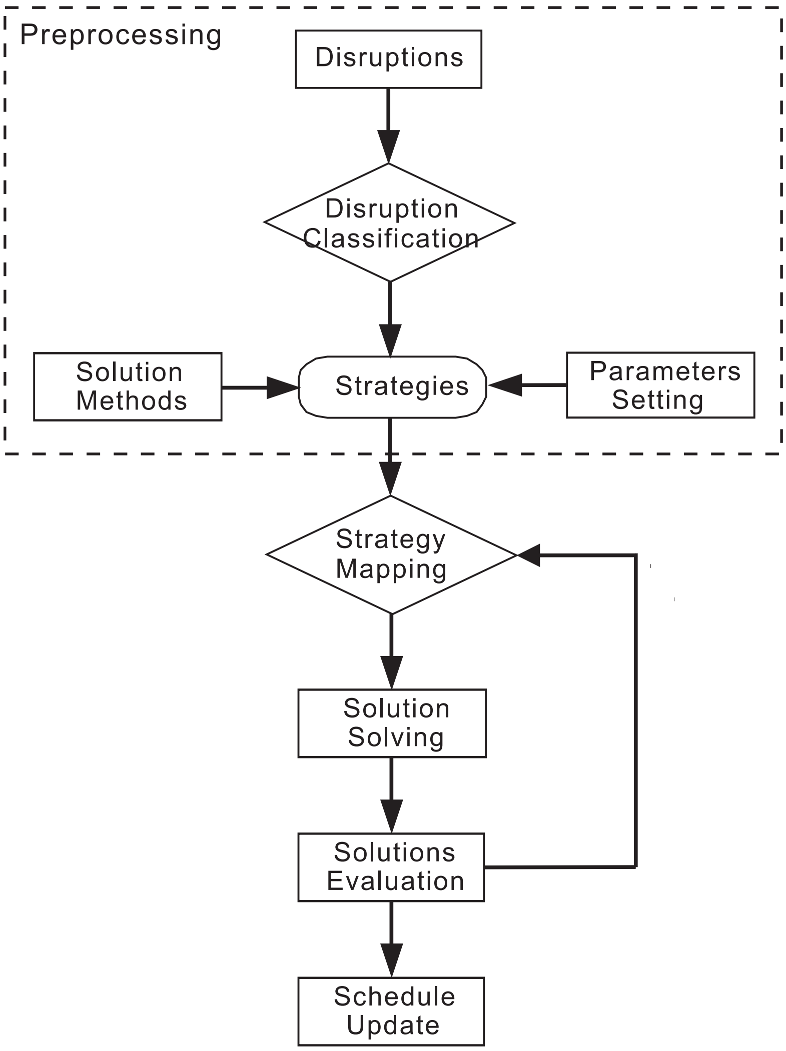


Figure 7.3: General crew recovery process flow

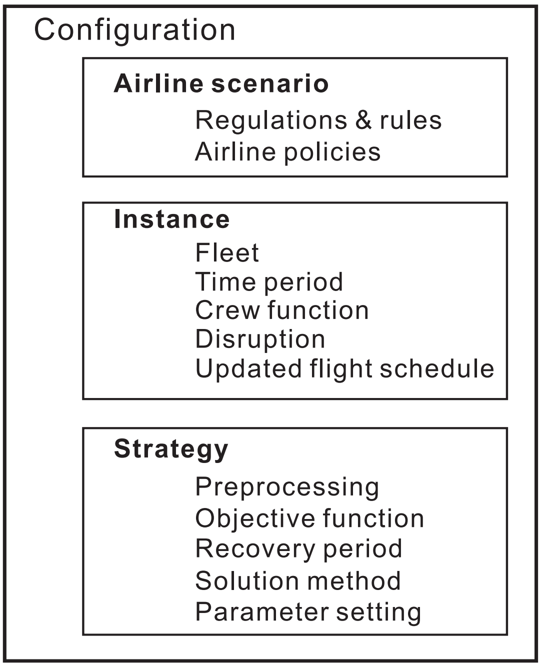


Figure 7.4: DSS configuration for the crew recovery process

setting and the system configuration, as other parts of the configuration are set up based on it. An instance is basically selected with corresponding parameters by answering the following questions: Which fleet tends to be planned; what time period is examined; which crew position is taken into consideration; what type of disruptions occur and how flight schedule is changed because of the disruptions. The parameters have tremendous effects on the size of the problem and the complexity for solving the problem. However, a strategy can be understood as the combination of several strategic decisions made to solve the problem differently and efficiently. As shown in the third part of the figure, five most important units play a significant role to form a strategy: preprocessing, objective function, recovery period, solution method and the setting of corresponding parameters.

After the preprocessing, a strategy mapping is carried out to find an appropriate strategy for the given disrupted situation. The problem is, therefore, solved with the specific method proposed in the strategy. One or more recovery solutions are finally produced and then evaluated. In the case that the generated solutions fail to prove applicable or useful, coordinators will repeat the strategy mapping procedure to choose another strategy and solve the problem again until an acceptable solution is found. The final step is to update and publish the crew schedule with respect to all changes made in the new schedule. The newly generated crew schedule is stored in a XML document, in which every change is marked with an attribute indicating a new assignment. A certain number of notifications have to be made in order to inform relevant crews about the changes. Ultimately, the crew schedule is recovered from disruptions through such crew recovery process flow. Within each step in the whole recovery process, the DSS provides dedicated tools, support, methods or techniques which are able to accomplish goals in a systematic way.

7.5 Summary

In this chapter, we propose a decision support system that covers the complete life cycle of the airline crew scheduling process, which is designed to meet the requirements for both planning and operational phase. We develop a general infrastructure for the crew management processes in which alternative solution methods, such as mathematical programming and heuristics, can be easily incorporated. Within the system, we suggest a dedicated crew recovery approach to handle unexpected disruptions. The system is able to single out one strategy, with which the examined problem can be solved more efficiently by customizing different solution methods. With such a system, we can cope with the damage of crew schedules more effectively than the approach with a single specific method.

Chapter 8

Conclusions and Future Research

In previous chapters, we have studied the airline crew recovery problem and proposed several solution methods, a strategy mapping scheme, as well as the prototype of a dedicated decision support system. This chapter concludes the thesis by highlighting our contributions to this particular problem, and suggesting some potential directions for future research.

During the initial involvement in the area of the airline crew scheduling, the emphasis of our research was placed on the two sub-problems — crew pairing and crew rostering — and the integration of both. After we studied the solution methods for crew scheduling in the planning phase, we observed another challenging task faced by airlines every day to handle disrupted situations in the operational phase, namely the airline crew recovery problem. Only recently this problem begins to attract researchers' attention, therefore relatively limited work has been done on this specific topic. The fast growth of the air traffic and the complexity of airlines' operational environment frequently causes disturbances to normal operations. Therefore, addressing efficient algorithms and supporting systems becomes more and more urgent. This motivates us to work on the problem of rescheduling crews during irregular operations and to develop a dedicated decision support system that incorporates and manages several algorithms.

Our first contribution provides a detailed description of the airline crew recovery problem, especially in the setting of most European tourist airlines. In Chapter 1 and 2, we elaborated on airlines' operation environment in which airlines generate their crew schedules and update them for the purpose of recovering from disruptions. The detailed problem structure was presented including involved resources, activities, constraints, cost structure, and so on. The crew recovery process in practice was also reported, in which several practical issues were discussed in the case that the process is carried out in a manual manner. Additionally, significant differences between the airline crew scheduling in the planning phase and the recovery process in the operational phase were explained, which led to the development of real-time solution methods for solving the recovery problem.

Because of the great similarity and the close relationship, we conducted a literature review on both the airline crew scheduling and rescheduling problem in Chapter 3. As the concepts and techniques developed for the former problem can be applied to the crew recovery problem, we provided a survey of previous work on both problems. It concluded the lack of research on the crew recovery problem in the setting of European airlines, and also showed the need to build a dedicated decision support system.

In Chapter 4, we modeled this task as two different set partitioning problems which differ from previous attempts since our approaches do not address the concept of pairing. In other words, crew rosters with the length of the recovery period are constructed directly from the flight level. Since it does not introduce any sub-problems, it allows us to achieve better global solutions. In addition to the direct solving approach, we designed and implemented a column generation approach to solve the problem in an implicit way. The subproblem of the column generation is solved by a constrained short path algorithm using dynamic programming. The column generation approach showed a dramatic improvement in terms of solution time compared with the direct solving method. Furthermore, the result has demonstrated the high possibility of finding an optimal solution because multiple ones exist in this particular type of problems.

The time restriction imposed by this particular problem motivated us to develop heuristics that can solve the problem in a short period of time. In Chapter 5 we presented two heuristic approaches including a GA based heuristic and a constructive algorithm. The constructive algorithm solves the problem in a simple and intuitive way, which leads to a fast solution approach but mostly finds solutions with limited quality. Therefore, we proposed a GA based approach in which a new two dimensional matrix representation is adopted and various heuristics are applied accounting for specific characteristics of the problem. Several variation operators have been implemented and tested. With the GA approach, we can solve very large problem instances in reasonable time and produce acceptable recovery solutions.

Due to the complexity of the examined problem, no one single method can solve all problem instances better than others. Therefore, in Chapter 6 we proposed a strategy mapping procedure using the AHP technique, by which we can single out an appropriate strategy based on the preliminary investigation of the given disruption.

Finally, in Chapter 7 we presented the prototype of a dedicated decision support system, in which all solution methods presented earlier and the strategy mapping procedure are incorporated. Coordinators of airlines can therefore recover the damaged crew schedule using a dedicated solution method selected by the strategy mapping procedure. Furthermore, the decision support system was merged with the approach for crew scheduling in the planning phase. Therefore, the resulting system is able to cover the entire life cycle of the airline crew scheduling process.

Although the column generation approach can solve most problem instances and mostly find an optimal solution, it cannot guarantee an optimal integer solution theoretically. Therefore, one further research direction is to embed it into a branch-and-bound procedure, which forms a branch-and-price approach (as studied in Barnhart et al., 1999a). However, further work on branching strategies and bounding techniques has to be conducted in order to reduce the total solution time, especially for large problem instances. Another further research direction is to extend the GA based heuristic by implementing more variation operators. One direction is to develop a similar crossover to the so called *conflicts-based crossover* proposed by Lewis and Paechter (2004) to solve the school timetabling problem. However, in our problem, we may randomly select a flight leg f and identify collections of building blocks that are applicable to common crew members. Therefore, this operator can potentially improve the average quality of the population over generations.

With the fast development of computer hardware, we may also consider other recovery processes into one single approach in future, namely aircraft and passenger recovery. By integrating the three approaches or at least increasing the interaction in between, airlines may handle disruptions even better in terms of cost saving, fast response, and less conflicts.

Bibliography

A. Abdelghany, G. Ekollu, R. Narasimhan, and K. Abdelghany. A proactive crew recovery decision support tool for commercial airlines during irregular operations. *Annals of Operations Research*, 127:309–331, 2004. 3.2.2

H. Achour, M. Gamache, and F. Soumis. Branch and cut at the subproblem level in a column generation approach: Application to the airline industry. Technical Report G-2003-34, GERAD, 2003. 3.1.2.1

Y. Ageeva. Approaches to incorporating robustness into airline scheduling. Master's thesis, Department of Electrical Engineering and Computer Science, MIT, 2000. 2

J. Alcaraz and C. Maroto. A robust genetic algorithm for resource allocation in project scheduling. *Annals of Operations Research*, 102:83–109, 2001. 5

P. Alefragis, P. Sanders, T. Takkula, and D. Wedelin. Parallel integer optimization for crew scheduling. *Annals of Operations Research*, 99(1):141–166, 2000. 3.1.1.2

R. Anbil, C. Barnhart, E. Johnson, and L. Hatay. A column generation technique for the long-haul crew assignment problem. In T. Ciriani and R. Leachman, editors, *Optimization in Industry II*, pages 7–24. John Wiley & Sons, 1994. 3.1.1.2

R. Anbil, J. J. Forrest, and W. R. Pulleyblank. Column generation and the airline crew pairing problem. *Documenta Mathematica*, Extra Volume ICM:677–686, 1998. In Proceedings of the International Congress of Mathematicians Berlin. 3.1.1.2

R. Anbil, E. Gelman, B. Patty, and R. Tanga. Recent advances in crew-pairing optimization at american airlines. *Interfaces*, 21:62–74, 1991. 3.1.1.2

R. Anbil, R. Tanga, and E. L. Johnson. A global approach to crew pairing optimization. *IBM System Journal*, 31(1):71–78, 1992. 3.1.1.2

E. Andersson, E. Housos, N. Kohl, and D. Wedelin. Crew pairing optimization. In G. Yu, editor, *Operations Research in the Airline Industry*. Kluwer Academic Publishing, 1997. 3.1.1.2

J. Antes. Structuring the process of airline scheduling. Technical report, University of Cologne, Germany, 1997. URL `http://www.informatik.uni-koeln.de/winfors/pub/ja-sor97-1.ps`. 2.1.1

J. P. Arabeyre, J. Fearnley, F. C. Steiger, and W. Teather. The airline crew scheduling problem: A survey. *Transportation Science*, 3(2):140–163, 1969. 3.1

E. Baker, L. Bodin, W. Finnegan, and R. Ponder. Efficient heuristic solutions to an airline crew scheduling problem. *AIIE Trans.*, 11(2):79–85, 1979. 3.1.1.2

J. F. Bard, G. Yu, and M. F. Argüello. Optimizing aircraft routings in response to grounding and delays. *IIE Transactions*, 33:931–947, 2001. 3.2.2

C. Barnhart, P. Belobaba, and A. R. Odoni. Applications of operations research in the air transport industry. *Transportation Science*, 37(4):368–391, 2003. 3

C. Barnhart and A. Cohn. Commissioned paper: Airline schedule planning: Accomplishments and opportunities. *Manufacturing & Service Operations Mgmt.*, 6(1):0003–0022, 2004. 3, 3.1.1.2

C. Barnhart, A. M. Cohn, E. L. Johnson, D. Klabjan, G. L. Nemhauser, and P. H. Vance. Airline crew scheduling. In Hall R.W., editor, *Handbook of Transportation Science*. Kluwer, 1999a. Updated version. 2.2.6, 3, 3.1.1.1, 3.1.1.1, 3.1.2.2, 3.2.2, 8

C. Barnhart, E. L. Johnson, R. Anbil, and L. Hatay. A column generation technique for the long-haul crew assignment problem. In T.A. Cirani and R.C.

Leachman, editors, *Optimization in industry*, volume 2, pages 7–23. Chichester: Wiley, 1994. 3.1.1.2, 3.1.2

C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. P. Savelsbergh, and P. H. Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46:316–329, 1998. 3.1.1.2

C. Barnhart, E. L. Johnson, G. L. Nemhauser, and P. H. Vance. Crew scheduling. In R.W. Hall, editor, *Handbook of Transportation Science*, pages 493–521. Kluwer Academic Publisher, Norwell, 1999b. 3.1, 3.1

J. Beasley and P. Chu. A genetic algorithm for the set covering problem. *European Journal of Operational Research*, 94:392–404, 1996. 5, 5.1.7

J. E. Beasley and B. Cao. A dynamic programming based algorithm for the crew scheduling problem. *Computers & Operations Research*, 25(7-8):567–682, 1998. 3.1

A. Bölte and U. W. Thonemann. Optimizing simulated annealing schedules with genetic programming. *European Journal of Operational Research*, 92(2):402–416, 1996. 5.1.6

J. Buhr. Four methods for monthly crew assignment — a comparison of efficiency. In *AGIFORS Symposium Proceedings*, volume 18, pages 403–430, 1978. 1b

E. R. Butchers, P. R. Day, A. P. Goldie, S. Miller, J. A. Meyer, D. M. Ryan, A. C. Scott, and C. A. Wallace. Optimized crew scheduling at air new zealand. *Interfaces*, 31(1):30–55, 2001. 4

J. Byrne. A preferential bidding system for technical aircrew. In *AGIFORS Symposium Proceedings*, volume 28, pages 87–99, 1988. 1c

K. W. Campbell, R. B. Durfee, and G. S. Hines. Fedex generates bidlines using simulated annealing. *Interfaces*, 27(2):1–16, 1997. 3.1.2.1

P. Cappanera and G. Gallo. On the airline crew rostering problem. Technical report, Dipartimento Di Informatica, University of Pisa, 2001. TR-01–08. 5

P. Cappanera and G. Gallo. A multicommodity flow approach to the crew rostering problem. *Operations Research*, 52(4):583–596, 2004. 5

A. Caprara, P. Toth, D. Vigo, and M. Fischetti. Modeling and solving the crew rostering problem. *Operations Research*, 46:820–830, 1998. 3.1

L. Cavique, C. Rego, and I. Themido. Subgraph ejection chains and tabu search for the crew scheduling problem. *Journal of the Operational Research Society*, 50:608–616, 1999. 3.1.1.2

S. Chebalov and D. Klabjan. Robust airline crew scheduling: Move-up crews. In *Proceedings of the 2002 NSF Design, Service, and Manufacturing Grantees and Research Conference*, 2002. 2

I. T. Christou, A. Zakarian, J. M. Liu, and H. Carter. A two-phase genetic algorithm for large-scale bidline-generation problems at delta air lines. *Interfaces*, 29(5):51–65, 1999. 3.1.2.1

P. Chu and J. Beasley. A genetic algorithm for the set partitioning problem. Technical report, Imperial College, The Management School, London, England, 1995. URL `http:/mscmga.ms.ic.ac.uk/pchu/pchu.html`. 5

L. W. Clarke, C. A. Hane, E. L. Johnson, and G. L. Nemhauser. Maintenance and crew considerations in fleet assignment. *Transportation Science*, 30(3): 249–260, 1996. 3.2.2

M. Clarke. Solving the problem of irregular airline operations, 1995. Presented at the INFORMS New Orleans Fall 1995 Meeting Airline Operations Planning and Control Section on Aviation Applications. 3.2.2

M. D. Clarke, L. Lettovský, and B. Smith. The development of the airline operations control center. In D. Jenkins, editor, *Handbook of Airline Economics*, pages 197–215. McGraw-Hill, 2002. 3.2, 3.2.2

M. D. D. Clarke. The airline schedule recovery problem, 1997. International Center for Air Transportation, Massachusetts Institute of Technology. 3.2.2

M. D. D. Clarke. Irregular airline operations: a review of the state-of-the-practice in airline operations control centers. *Journal of Air Transport Management*, 4 (2):67–76, 1998. 3.2.2

A. Cohn and C. Barnhart. Improving crew scheduling by incorporating key management routing decisions. *Operations Research*, 51(3):387–396, 2003. 3.1.3

T. G. Crainic and J. M. Rousseau. The column generation principle and the airline crew scheduling problem. *INFOR*, 25(2):136–151, 1987. 3.1.1.2

H. Dawid, J. König, and C. Strauss. An enhanced rostering model for airline crews. *Computers and Operations Research*, 28(7), 2001. 3.1.2.2, 4

P. R. Day and D. M. Ryan. Flight attendant rostering for short haul airline operations. *Operations Research*, 45(5), 1997. 2.1.1

G. Desaulniers, J. Desrosiers, I. Ioachim, M. Solomon, and F. Soumis. A unified framework for deterministic time constrained vehicle routing and crew scheduling problems. In T. Crainic and G. Laporte, editors, *Fleet Management and Logistics*, pages 57–93. Kluwer Publishing Company, 1998. 3.1.1.2

M. Desrochers and F. Soumis. A column generation approach to the urban transit crew scheduling problem. *Transportation Science*, 23(1):1–13, 1989. 3.1.1.2, 4.3.1

J. Desrosiers, Y. Dumas, M. Desrochers, F. Soumis, B. Sanso, and P. Trudeau. A breakthrough in airline crew scheduling. Technical Report G-91-11, GERAD, 1991. 3.1.1.2

J. Desrosiers, Y. Dumas, M. M. Solomon, and F. Soumis. Time constrained routing and scheduling. In M. Ball, T. Magnanti, C. Monma, and G. Newhauser, editors, *Network Routing*, volume 8 of *Handbooks in Operations Research and Management Science*, pages 35–140. Elsevier, Amsterdam, 1995. 3.1.1.2, 4.3.3.3

J. Desrosiers and M. E. Lübbecke. A primer in column generation. Technical report, Technische Universität Berlin, 2003. URL `http://www.math.tu-berlin.de/coga/publications/techreports/2003/Report-048-2003.html`. 3.1.1.2, 4.3.3

T. G. Dias, J. P. Sousa, and J. F. Cunha. A genetic algorithm for the bus driver scheduling problem. In *Proceedings of 4th Metaheuristics International Conference (MIC2001)*, pages 35–40, 2001. 3.1, 5

K. Doerner, G. Kotsis, and C. Strauss. Rosterbuilder - an architecture for an integrated airline rostering framework. *Information Technology & Tourism*, 6 (1):69–83, 2003. 3.1.2.1

W. El Moudani, C. Cosenza, M. de Coligny, and F. Mora-Camino. A bi-criterion approach for the airline crew rostering problem. In E. Zitzler, K. Deb, L. Thiele, C. A. Coello Coello, and D. Corne, editors, *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization (EMO 2001)*, pages 486–500, Berlin, 2001. Springer-Verlag. 3, 5.1.1.1

T. Emden-Weinert and M. Proksch. Best practice simulated annealing for the airline crew scheduling problem. *Journal of Heuristics*, 5(4):419–436, 1999. ISSN 1381-1231. 3.1.1.2

A. T. Ernst, H. Jiang, M. Krishnamoorthy, and D. Sier. Staff scheduling and rostering: A review of applications, methods and models. *European Journal of Operational Research*, 153(1):3–27, 2004. 3.1.1.2

M. M. Etschmaier and D. F. X. Mathaisel. Airline scheduling: An overview. *Transportation Science*, 19(2):127–128, 1985. 3

Eurocontrol. Delays to air transport in europe annual report 2002 air transport reports, 2002. URL `http://www.eurocontrol.int`.

Eurocontrol. Delays to air transport in europe annual report 2003 air transport reports, 2003. URL `http://www.eurocontrol.int`. 1.2, 2.2.4

European Regions Airline Association. Performance (January to December 2003) air transport reports, 2003. URL `http://www.eraa.org`. 1.2, 2.2.4

T. Fahle, U. Junker, S. Karisch, N. Kohl, M. Sellmann, and B. Vaaben. Constraint programming based column generation for crew assignment. *Journal of Heuristics*, 8:59–81, 2002. 6

J.A. Filar, P. Manyem, and K. White. How airlines and airports recover from schedule perturbations: A survey. *Annals of Operations Research*, 108, 2001. 3.2.2

R. Freling, D. Huisman, and A. P. M. Wagelmans. Models and algorithms for integration of vehicle and crew scheduling, 2000. Econometric Institute Report EI2000-10/A, Econometric Institute, Erasmus University Rotterdam, Netherlands. 3.1.3

R. Freling, R.M. Lentink, and A. P. M. Wagelmans. A decision support system for crew planning in passenger transportation using a flexible branch-and-price algorithm. Econometric Institute Report EI 2001-29, Erasmus University Rotterdam, Econometric Institute, 2001. 3.1, 3.1.1.2, 6

Z. Fu, B. L. Golden, S. Lele, S. Raghavan, and E. Wasil. A genetic algorithm-based approach for building accurate decision trees. *INFORMS Journal on Computing*, 15(1):3–22, 2003. 5

M. Gamache and F. Soumis. A method for optimally solving the rostering problem. In G. Yu, editor, *Operations Research in the Airline Industry*, pages 124–159. Kluwer Academic Publishing, 1998. 3.1.2.2, 6

M. Gamache, F. Soumis, G. Marquis, and J. Desrosiers. A column generation approach for large-scale aircrew rostering problems. *Operations Research*, 47 (2):247–263, 1999. 3.1.2.3, 6

M. Gamache, F. Soumis, D. Villeneuve, J. Desrosiers, and É. Gélinas. The preferential bidding system at air canada. *Transportation Science*, 32(3):246–255, 1998. 3.1.2.1

M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, New York, 1979. 4.3.2, 4.3.3.2

I. Gershkoff. Optimizing flight crew schedules. *Interfaces*, 19(4):29–43, 1989. 2.2.6

C. Giafferri, J. P. Hamon, and J. G. Lengline. Automatic monthly assignment of medium-haul cabin crew. In *AGIFORS Symposium Proceedings*, volume 22, pages 69–95, 1982. 1d

W. Glanert. A timetable approach to the assignment of pilots to rotations. In *AGIFORS Symposium Proceedings*, volume 24, pages 369–391, 1984. 1a

R. Gopalan and K. T. Talluri. Mathematical models in airline schedule planning: A survey. *Annals of Operations Research*, 76(0):155–185, 1998. 3

Y. Guo, T. Mellouli, L. Suhl, and M. P. Thiel. A partially integrated airline crew scheduling approach with time-dependent crew capacities and multiple home bases. Technical Report WP0303, DS&OR Lab., University of Paderborn, Germany, 2003. Accepted by European Journal of Operational Research. 3.1.1.2, 3.1.3, 5.2, 5.2.1

S. Hartmann. A competitive genetic algorithm for resource-constrained project scheduling. *Naval Research Logistics*, 45:733–750, 1998. 5.1.1.1

A. Hertz and M. Widmer. Guidelines for the use of meta-heuristics in combinatorial optimization. *European Journal of Operational Research*, 151:247–252, 2003. 5

C. A. Hjorring and J. Hansen. Column generation with a rule modelling language for airline crew pairing. In *Proceedings of the 34th Annual Conference of the Operational Research Society of New Zealand*, pages 133–142, 1999. 3.1.1.2

C. A. Hjorring, S. E. Karisch, and N. Kohl. Carmen systems' recent advances in crew scheduling. Technical report, Carmen Systems AB, 2000. URL `http://www.carmen.se`. 3.1.1.2

K. L. Hoffman and M. Padberg. Solving airline crew scheduling problems by branch-and-cut. *Management Science*, 39(6):657–682, 1993. 3.1.1.2

J. Holland. *Adaption in natural and artificial systems*. The University of Michigan Press, 1975. 5, 5.1.7

ILOG. *Cplex v8.0 User's Manual*. ILOG, France, 2002. 3.1.1.2, 4.3.2, 4.4

M. E. Irrgang. Airline irregular operations. In D. Jenkins and C.P. Ray, editors, *Handbook of airline economics*, pages 349–365. McGrawHill Aviation Week Group, New York, 1st edition, 1995. 3.2.2

A. I. Z. Jarrah and J. T. Diamond. The problem of generating crew bidlines. *Interfaces*, 27(4):49–64, 1997. 3.1.2.1

U. Junker, S. Karisch, N. Kohl, B. Vaaben, T. Fahle, and M. Sellmann. A framework for constraint programming based column generation. In *Proceedings of CP 1999*, pages 261–274, 1999. 6

S. E. Karisch. Carmen crew pairing and rostering: Technology and applications. Technical report, Carmen Systems AB, 2003. URL `http://www.carmen.se/`. CRTR-2003-1. 3.1.1.2

S. Kerati, W. EL Moudani, M. de Coligny, and F. Mora-Camino. A heuristic genetic algorithm approach for the airline crew scheduling problem, 2002. URL `http://www.lifl.fr/PM2O/Reunions/04112002/kerati.pdf`. 3

D. Klabjan, E. L. Johnson, G. L. Nemhauser, E. Gelman, and S. Ramaswamy. Solving large airline crew scheduling problems: Random pairing generation and strong branching. *Comput. Optim. Appl.*, 20(1):73–91, 2001. ISSN 0926-6003. 3.1.1.1

D. Klabjan, E. L. Johnson, G. L. Nemhauser, E. Gelman, and S. Ramaswamy. Airline crew scheduling with time windows and plane-count constraints. *Transportation Science*, 36(3):337–348, 2002. 3.1.3

D. Klabjan and K. Schwan. Airline crew pairing generation in parallel. In *Proceedings of the Tenth SIAM Conference on Parallel*, 2000. 2, 3.1.1.2

N. Kohl and S. E. Karisch. Airline crew rostering: problem types, modeling, and optimization. Carmens Systems, 2002. URL `http://www.carmen.se/`. 3rd. revised version (July 2003). 3.1

N. Kohl and S. E. Karisch. Airline crew rostering: Problem types, modeling, and optimization. *Annals of Operations Research*, 127:223–257, 2004. 3.1.2.1, 3.1.2.3

H. König and C. Strauss. Rostering-integrated services and crew efficiency. *Information Technology and Tourism*, 3(1):27–39, 2000. 3.1.2.2

L. Kwok and L. Wu. Development of an expert system in cabin crew pattern generation. *International Journal of Expert Systems*, 9:445–464, 1996. 2.1.1

S. Lavoie, M. Minoux, and E. Odier. A new approach for crew pairing problems by column generation with an application to air transportation. *European Journal of Operational Research*, 35(1):45–58, 1988. 3.1.1.2

L. Lettovský, E. L. Johnson, and G. L. Nemhauser. Airline crew recovery. *Transportation Science*, 34(4):337–348, 2000. 3.2.2

D. Levine. Application of a hybrid genetic algorithm to airline crew scheduling, 1996. URL `citeseer.ist.psu.edu/levine96application.html`. 3, 5.1.5, 5.1.7

R. Lewis and B. Paechter. New crossover operators for timetabling with evolutionary algorithms. In *Proceedings of 5th International Conference on Recent Advances in Soft Computing (RASC 2004)*, pages 189–194, 2004. 8

M. Løve, K. R. Sørensen, J. Larsen, and J. Clausen. Using heuristics to solve the dedicated aircraft recovery problem. Technical report, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby, 2001. URL `http://www.imm.dtu.dk/pubdb/p.php?138`. 3.2.2

M. E. Lübbecke and J. Desrosiers. Selected topics in column generation. Technical report, Technische Universität Berlin, 2004. URL `http://www.math.tu-berlin.de/coga/publications/techreports/2004/Report-008-2004.html`. 4.3.3

P. Lučić and D. Teodorović. Simulated annealing for the multi-objective aircrew rostering problem. *Transportation Research Part A*, 33:19–45, 1999. 2

F. Marchettini. Automatic monthly cabin crew rostering procedure. In *AGIFORS Symposium Proceedings*, volume 20, pages 23–59, 1980. 1a

E. Marchiori and A. Steenbeek. An evolutionary algorithm for large scale set covering problems with application to airline crew scheduling. In S. Cagnoni et al., editor, *Real-World Applications of Evolutionary Computing: EvoWorkshops 2000: EvoIASP, EvoSCONDI, EvoTEL, EvoSTIM, EvoRob, and EvoFlight, Edinburgh, Scotland, UK*, pages 367–381. Springer-Verlag Heidelberg, 2000. 3

T. Mellouli. A network flow approach to crew scheduling based on an analogy to a train/aircraft maintenance routing problem. In S. Voß and J.R. Daduna, editors, *Computer-Aided Scheduling of Public Transport*, pages 91–120. Springer, Berlin, 2001. 3.1.1.2

T. Mellouli. Scheduling and routing processes in public transport systems. Habilitation Thesis, University of Paderborn, Germany, 2003. 3.1, 3.1.1.2

Z. Michalewicz and D. B. Fogel. *How to Solve It: Modern Heuristics*. Springer-Verlag, 2000. 5.1.1.1, 5.1.6

A. Mingozzi, M. A. Boschetti, S. Ricciardelli, and L. Bianco. A set partitioning approach to the crew scheduling problem. *Operations Research*, 47(6):873–888, 1999. 3.1.1.1

M. Minoux. Column generation techniques in combinatorial optimization: A new application to crew pairing problems. In *Proceedings of XXIVth AGIFORS Symposium*, 1984. 3.1.1.2

R. Moore, J. Evans, and H. Noo. Computerized tailored blocking. In *AGIFORS Symposium Proceedings*, volume 18, pages 343–361, 1978. 1c

B. Nicoletti. Automatic crew rostering. *Transportation Science*, 9(1):33–42, 1975. 1b

R. Nissen. *Airline crew rescheduling*. PhD thesis, Christian-Albrechts-University Kiel, 2003. 3.2.2

C. R. Reeves. *Modern heuristic techniques for combinatorial problems*. John Wiley & Sons, 1993. ISBN 0-470-22079-1. 5.1.2

J. Richardson, M. palmer, G. Liepins, and M. Hilliard. Some guidlines for genetic algorithms with penalty functions. In *Proceedings of the Third International Conference on Genetic Algorithms*, San Mateo, 1989. Morgan Kaufmann. 5.1.4

J. M. Rosenberger, E. L. Johnson, and G. L. Nemhauser. Rerouting aircraft for airline recovery. *Transportation Science*, 37(4):408–421, 2003. 3.2.2

J. M. Rosenberger, A. J. Schaefer, D. Goldsman, E. L. Johnson, A. J. Kleywegt, and G. L. Nemhauser. A stochastic model of airline operations. *Transportation Science*, 36(4):357–377, 2002. 3.2.2

J. Rubin. Technique for the solution of massive set covering problems, with application to airline crew scheduling. *Transp. Sci.*, 7(1):34–48, 1973. 3.1

R. A. Rushmeier, K. L. Hoffman, and M. Padberg. Recent advances in exact optimization of airline scheduling problems. Technical report, George Mason University, 1995. 3, 4.3.3

D. M. Ryan. The solution of massive generalized set partitioning problems in aircrew rostering. *Journal of the Operational Research Society*, 43(5):459–567, 1992. 4

D. M. Ryan and J. C. Falkner. On the integer properties of scheduling set partitioning models. *European Journal of Operational Research*, 35:442–456, 1988. 4

T. L. Saaty. *The Analytic Hierarchy Process*. McGraw-Hill, New York, 1980. 6.2.1

P. Sanders, T. Takkula, and D. Wedelin. High performance integer optimization for crew scheduling. In *HPCN Europe 1999*, pages 3–12, 1999. 3.1.1.2

D. Sarra. The automatic assignment model. In *AGIFORS Symposium Proceedings*, volume 28, pages 23–37, 1988. 1b

M. S. Silver. *Systems that support decision makers: description and analysis*. John Wiley & Sons, 1991. 7.1

M. Song, G. Wei, and G. Yu. A decision support framework for crew management during airline irregular operations. In G. Yu, editor, *Operations Research in the Airline Industry*, pages 260–286. Kluwer Academic Publishing, 1998. 3.2.2

G. Stojković, F. Soumis, J. Desrosiers, and M. M. Solomon. An optimization model for a real-time flight scheduling problem. *Transportation Research Part A: Policy and Practice*, 36(9):779–788, 2002. 3.2.2

M. Stojković and F. Soumis. An optimization model for the simultaneous operational flight and pilot scheduling problem. *Management Science*, 47(9): 1290–1305, 2001. 3.2.2

M. Stojković, F. Soumis, and J. Desrosiers. The operational airline crew scheduling problem. *Transportation Science*, 32(3):232–245, 1998. 3.2.2

L. Suhl. *Computer-Aided Scheduling—An Airline Perspective*. Deutscher Universitäts-Verlag (DUV), Wiesbaden, 1995. 1.1, 1, 3.1

U. H. Suhl. MOPS: A Mathematical OPtimization System. *European Journal of Operational Research*, 72:312–322, 1994. 3.1.1.2

U. H. Suhl. MOPS - Mathematical OPtimization System. *OR News, Nr. 8, 2000*, pages 11–16, 2000. 3.1.1.2

B. G. Thengvall, J. F. Bard, and G. Yu. A bundle algorithm approach for the aircraft schedule recovery problem during hub closures. *Transportation Science*, 37(4):392–407, 2003. 3.2.2

M. P. Thiel. Team-oriented airline crew rostering for cockpit personnel. Technical Report WP0406, DS&OR Lab., University of Paderborn, Germany, 2004. Presented at CASPT 2004, San Diego, CA, the U.S. 3.1.2.2

D. Thierens and D. E. Goldberg. Convergence models of genetic algorithm selection schemes. In *PPSN III: Proceedings of the International Conference on Evolutionary Computation. The Third Conference on Parallel Problem Solving from Nature*, pages 119–129, London, UK, 1994. Springer-Verlag. 5.1.6

H. Timucin Ozdemir and C. K. Mohan. Graga: a graph based genetic algorithm for airline crew scheduling. In *11th IEEE International Conference on Tools with Artificial Intelligence*, pages 27–28, 1999. 3.1.1.2, 3

G. A. Tingley. Still another solution method for the monthly aircew assignment problem. In *AGIFORS Symposium Proceedings*, volume 19, pages 143–203, 1979. 1b

E. Turban and J. E. Aroson. *Decision Support Systems And Intelligent Systems*. Prentice-Hall, Inc., Upper Saddle River, New Jersey, sixth edition, 2000. 7.1

P. H. Vance, C. Barnhart, E. L. Johnson, and G. L. Nemhauser. Airline crew scheduling: A new formulation and decomposition algorithm. *Operations Research*, 45:188–200, 1997. 3.1.1.1, 3.1.1.2

M. B. Wall. *A Genetic Algorithm for Resource-Constrained Scheduling*. PhD thesis, Department of Mechanical Engineering, MIT, 1996. 5

D. Wedelin. An algorithm for large scale 0-1 integer programming with application to airline crew scheduling. *Annals of Operations Research*, 57:283–301, 1995. 3.1.1.1

G. Wei, G. Yu, and M. Song. Optimization model and algorithm for crew management during airline irregular operations. *Journal of Combinatorial Optimization*, 1(3):305–321, 1997. 2.4, 3.2, 3.2.1, 3.2.2, 4.2.1

W. E. Wilhelm. A technical review of column generation in integer programming. *Optimization and Engineering*, 2(2):159–200, 2001. 4.3.3

C.-L. Wu. The influence of operational uncertainties on airline schedule planning and punctuality control issues. In *Proceedings of CAITR-2003*, Adelaide, 2003. 1.2

Z. Xu and S. J. Louis. Genetic algorithms for open shop scheduling and rescheduling. In *Proceedings of the ISCA 11th International Conference on Computers and Their Applications*, pages 99–102, 1996. 5

S. Yan and C. G. Lin. Airline scheduling for the temporary closure of airports. *Transportation Science*, 31(1):72–83, 1997. 3.2.2

S. Yan and Y. P. Tu. A network model for airline cabin crew scheduling. *European Journal of Operational Research*, 140(3):531–540, 2002. 3.1.1.2

G. Yu, editor. *Operations Research in the Airline Industry*, volume 9 of *Operations Research and Management Science*. Kluwer Academic Publishers, 1998. 3

G. Yu, M. Argüello, G. Song, S. M. McCowan, and A. White. A new era for crew recovery at continental airlines. *Interfaces*, 33(1):5–22, 2003. 3.2, 3.2.2, 7.2

T. H. Yunes, A. V. Moura, and C. C. de Souza. Solving large scale crew scheduling problems with constraint programming and integer programming. Technical Report IC-99-19, Institute of Computing, University of Campinns, Brazil, 1999. URL `http://goa,pos.dcc.unicamp.br/otimo/published.html`. 6

T. H. Yunes, A. V. Moura, and C. C. de Souza. Solving very large crew scheduling problems to optimality. In *Proceedings of the 2000 ACM Symposium on Applied Computing*, pages 446–451, 2000. 3.1, 6

T. H. Yunes, A. V. Moura, and C. C. de Souza. Hybrid column generation approaches for solving real world crew management problems. In *Proceedings of the 27th Conferencia Latinoamericana de Informatica (CLEI 2001), Venezuela*, pages 24–28, 2001. 6