INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI

A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor MI 48106-1346 USA
313/761-4700 800/521-0600

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

A Decision Support Framework for Airline Crew Management During Irregular Operations

Approved by
Dissertation Committee:

Gang Yu, Supervisor

Jonathan Bard

Leon Lasdon

Patrick Jaillet

Soren Nielsen

A Decision Support Framework for Airline Crew Management During Irregular Operations

by

Guo Wei, B.S., M.A.

Dissertation

Presented to the Faculty of the Graduate School of
The University of Texas at Austin
in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

The University of Texas at Austin
May, 1997

UMI Number: 9803061

UMI
300 North Zeeb Road
Ann Arbor, MI 48103

Dedication

Dedicated to my parents, Jia Liquan and Wei Pingde, who had given and scarified everything they had to their son and their son's education. Without their love and support, I would not be able to finish this dissertation and accomplish whatever I have today.

Acknowledgements

The author wishes to thank his committee members for the many comments they have made. Special thanks go to my supervisor, Professor Gang Yu, who has inspired my interest in this fascinating field of operations research and provided invaluable guidance and feedback throughout the years I have been at the Graduate School of Business as a doctoral student. I owe much of my professional growth up to now to him.

I would also like to express my deep appreciation for Professor Patrick Jaillet, who not only sits on my dissertation committee, but also has been a constant source of encouragement and help. He had also played a very important role in introducing me into the world of management science.

Finally, during the writing of this dissertation, I have had the opportunity to discuss many aspects of airline operations with various people in different departments of Continental Airlines. They have helped me understand the airline industry and particularly irregular operations. I have also worked with many people from a number of divisions of EDS, including EDS airline operation in Los Angeles, EDS transportation division in Miami, EDS Continental account in Houston and EDS R&D in Austin. I have benefited immensely from working with them one way or the other. These people are just too numerous to list their names here. Much of the current dissertation is the result of discussions with them and I feel deeply indebted to them.

A Decision Support Framework for Airline Crew Management During Irregular Operations

Publication No.____________

Guo Wei, Ph.D.

The University of Texas at Austin, 1997

Supervisor: Gang Yu

Airline irregular operations are airline operations under irregular situations, which usually means some disturbances or perturbations have happened or are expected to happen in such a manner that they have disrupted or will disrupt the normal operations of the airline. At major airlines, crew management during irregular operations is a very inefficient process due to the complicated crew schedule, restrictive crew legalities and work rules. Very often, the fact that no appropriate solution is found in a timely manner results in flight cancellations and delays. The problem is also called crew pairing repair. Solving crew pairing repair problem quickly is critical to airline operations and will have direct impact to the airline's revenue. This dissertation discusses the crew management issues during irregular operations. A model is developed for the crew pairing repair problem. Both traditional OR algorithm and some heuristic algorithms are devised to solve the problem. It is demonstrated that the heuristic algorithms are far more flexible and are the preferred method. Using real data from an airline, the algorithms are shown efficient enough to solve large-scale crew pairing repair problem in real-time. In addition, future developments of the algorithms are also discussed. This dissertation is the first effort to use sophisticated model and algorithms to solve crew management problem during irregular operations.

Table of Contents

List of Tables

List of Figures

Chapter 1 INTRODUCTION

In this chapter, we give an overview of airline operations in general and airline irregular operations in particular. We discuss the various stages involved in airline operations, the kinds of problems that arise. We also discuss, briefly, the role operations research played in airline operations as well as some typical problems that can be solved by operations research methodologies. Definition and description of airline irregular operations are also provided. The purpose of this chapter is to provide an overall picture of airline operations without going into details the various problems and processes in airline operations.

1.1 AIRLINE OPERATIONS OVERVIEW

Operating a large, modern airline is an extremely complicated task. In the United States, the world's largest air transport market, most major airlines operate on the so-called hub-and-spoke system. In a hub-and-spoke system, an airline dominates a few strategically-located airports—called hubs—where it usually operates up to 70% of all the flights in the airports. Any two non-hub airports served by the airline can be connected by flights between the two airports and a hub—called spokes. The advantage of the networked, hub-and-spoke system over the linear, point-to-point system is that an airline can cover more markets with much fewer flights than would be required under a point-to-point system. There is also the marketing advantage for an airline operating under hub-and-spoke system. Because of its dominance in a hub, the airline can pick most of the passengers originating or passing through the hub. They can also send a passenger almost to any airport. These advantages, however, come at a price. The complexities of a network make it a very challenging task to build an airlines flight schedule, crew assignment schedule and maintenance schedule. These schedules are changed regularly in response to market changes, competition, or because of company policy and government regulation changes. All these factors make a difficult job even more challenging. It is a well known fact that major airlines invest tremendous capital and manpower into building these schedules.

Since the Airline Deregulation Act in 1978, airline industry in the U.S. has witnessed dramatic changes in its operating environment. Ruthless competition among major airlines and challenges from low-cost, smaller but nimbler airlines as well as changing customer expectations have put a downward pressure on revenues of major airlines. In response, airline industry has employed various marketing innovations such as frequent flyer program, yield management and has fiercely cut operating costs in all aspects.

Like many other industries, airline operations involve three different stages: planning, scheduling and control. These problems usually deal with deploying scarce resources (aircraft, crew) among competing activities (flights, routes) with the objective of optimization of some targets (e.g., revenue or profit maximization, cost minimization). In so doing, airline must also consider a host of operating constraints, including government regulations, company policies, weather patterns and equipment limitations.

Planning

Airline operation planning is about relatively long term, global and strategic decisions. It is usually market-driven. At this stage, an airline assesses the market demands, competition and the company resource capacity (aircraft, crew and other resources) to plan its product or service mix (flight schedules) in such a way so as to maximize its revenue or profit under the constraint of resource availability. The planning stage is necessarily done in aggregate and general terms, and is characterized by imprecise data with the help of experience and insight. It is proactive in nature.

For an airline, its sole product is its flight schedule. Thus the planning stage is centered around building an initial and rough flight schedule based on such strategic decisions as which markets to serve and at what level. This stage involves carefully analyzing these markets, including passenger traffic, competitive schedules as well as their potential profitability.

Scheduling

Scheduling, on the other hand, is a more tactical decision. Based on the decisions made in the planning stage, the objective of scheduling is the sequencing or ordering of activities (flights, crew staffing, maintenance services) and the assignment of resources (aircraft, crew, maintenance base and personnel) to such activities subject to availability and qualification of the resources, regulatory and company policies, such as aircraft maintenance requirements and crew legalities set by FAA (Federal Aviation Administration) and union contracts. The outcome of the scheduling stage is a feasible assignment mapping resources and activities. Compared with the planning problems, the objective of scheduling is usually to minimize cost while finding such an assignment.

For airlines, among the most important scheduling decisions are flight scheduling, crew scheduling and maintenance scheduling. For major airlines, which usually serve hundreds of airports and daily departure flights in the thousands, operate hundreds of aircraft of different types, and have thousands or even tens of thousands crew, scheduling is an enormously complex undertaking.

Control

Operational control, as its name suggests, is about control of operation. The goal of the airline operational control is to make sure the smooth execution of the plan and the various schedules made in the previous two stages. During the operation of an airline, unexpected events, such as weather, mechanical problems, crew shortage, general strikes, security problems and air traffic control (ATC) problems, can disrupt the plan and schedules. The airline therefore needs to carefully monitor the status of the operation on a real-time basis and make contingency plan when problems arise. For the major airlines operating under hub-and-spoke system, because of the way the system and schedules are built and the complexity of the schedules, problems in one area can quickly propagate to other areas and thus have down-line impact, especially when such problems happen in a hub, where large number of flights are concentrated, the impact could be very severe and global. The economic loss could be in the range of millions of or even tens of millions of dollars in the event of a major perturbation.

In most major airlines, there is usually a system control center or a similarly named unit which serves as the nerve center of the airline operation, not unlike the command, control and communication center during a war. The center is usually organized along the line of equipment types or fleet types due to the different qualifications and maintenance requirements. Within each equipment type, an operations manager takes overall control and the operations manager is paired with a crew coordinator who is responsible for providing information on crew duty limitations which aid the operations manager during decision making process. The crew coordinator schedules, tracks, and reschedules crew within the designated fleet type. The operations manager may also pair up with a customer service coordinator who advises the operations manager of customer impact and recommends alternatives during irregular operation (see next section for discussion on irregular operations). Other important positions inside the system control center may include aircraft dispatcher, maintenance control, etc.

Because of the dynamic and information-intensive nature of airline operations control, real-time information collection, storage and display are critical. This usually requires a modern communication and computing infrastructure. Also, strong communication and close coordination among different positions in the control center are essential to the decision-making process. At present, in most airlines, battle-tested experiences and rules of thumb are still the main weapons to tackle such a mission-critical task. The process is largely manual-driven and through oral communication. Therefore it is a slow process.

1.2 APPLICATIONS OF OPERATIONS RESEARCH IN AIRLINE OPERATIONS

Based on the previous discussion, it is not surprising that airline operations are a natural and rich field for applications of operations research. In fact, it is hard to think of any other industry in which operations research has found more applications and had larger impact. Because of the scope and complexity of airline operations, OR has played a critical role in almost every aspect of airline operations, such as network design, planning, scheduling and more recently, operational control. The fiercely competitive environment since the deregulation of airline industry has forced the airlines to cut cost relentlessly and improve their operational efficiency in order to survive. It is therefore no surprise that all major airlines have maintained a group of significant number of OR professionals.

In the following, a brief overview of applications of operations research in different stages of airline operations is given. It is not, however, meant to be a comprehensive review of all the researches that have been done in the field. The purpose here is to give an overall picture of OR applications in airline operations and to have an understanding of the kind of problems and their characteristics. It should also serve as a background information for the rest of this dissertation.

Planning

As has been discussed above, planning problems are usually long-term and aggregate. The data are static and may not be very accurate. The tools employed in solving planning problems are usually large-scale optimization-based models and corresponding algorithms.

A well-known planning problem is the network design problem. The purpose of solving the problem is to find an optimal solution for the hub-and-spoke network in such a manner that the airline can serve its targeted markets and deploy its resources most effectively and efficiently.

Yield-management, which was first employed by the airline industry and is now spread to many other industries, is another area where operations research has found its role. Using the large amount of marketing data collected from their powerful Computer Reservations System (CRS), major airlines have been trying to increase their revenue by manipulating the demands through creating different classes of services within the same flight and placing different requirements for different fare classes. The requirements of advance booking and/or Saturday night stay have effectively segmented the market into, for instance, business travelers and leisure travelers. Then, using advanced OR methodologies of forecasting and optimization and powerful computer systems, airlines can allocate hundreds of thousands seat inventory to thousands of different fare classes on a daily basis. Yield-management has been proved to generate major carriers additional revenues in hundreds of million dollars annually.

Due to the imprecise data input and the predictive nature of planning problems, it is very important for the OR-based models to have the what-if analysis capability to help management plan for different scenarios. The concept and methods of sensitivity analysis are therefore instrumental here.

Scheduling

Scheduling has long been an area of operations research where most of the problems are very hard and computationally intractable. Except for maybe a few simple cases, most scheduling problems are NP-hard problems. Thus, for almost all practical scheduling problems with realistic sizes, the only hope is to find some good, feasible solutions. The solution methods are usually combination of optimization-based models and some heuristics or rules-of-thumb that take advantage of the special characteristics of the problem at hand.

Among the more important scheduling problems in airline operations are flight scheduling and crew scheduling problems. Based on the strategic and market-driven decisions made during the planning stage, flight scheduling makes the decisions on cities to be served, frequency of the flights and the equipment types. Since an airline's sole product is its flight schedule, these decisions have direct impact on the company's revenue. One of the major flight scheduling problem is the Fleet Assignment Problem (FAP). This problem assigns an airline's fleet types to its flight schedule or flight legs. The basic trade-off here is that if the airline uses too small a plane, it will not be able to carry all the passengers, resulting in a loss of revenue, while if it uses too large a plane, it will incur a larger cost of larger plane taking off with empty seats. The objective function therefore could be cost minimization, profit maximization or minimization of number of planes used to fly the schedule to allow flexibility. The constraints may include the size of a particular fleet type, balance of aircraft (flow conservation), maintenance requirements and other constraints. The problem is generally formulated and solved as a large-scale mixed-integer program.[1]

Crew scheduling is another difficult problem for large airline operating under hub-and-spoke network system. Crew costs are the second highest component of direct operating cost after fuel.[2] A good, efficient crew schedule can contribute directly to the airlines' bottom line. The challenge of building a good crew schedule is how to assign thousands or even tens of thousands crew members to flights and at the same time to satisfy a complex set of regulatory rules established by Federal Aviation Administration seeking to reduce crew fatigue, union contract and company policies. The problem is further complicated by the different qualifications of the crew members and since crew members are human, they are considerably more complicated than scheduling other resources such as aircraft and gates. The crew scheduling problem is generally formulated as a set-partitioning problem, a classic NP-hard problem. Significant progress has been made in solving crew scheduling problem which results in huge cost saving for major airlines, [2, 3] but an optimal solution is still elusive given the size and difficulty of the problem. Thus there is still great potential in cost reduction from crew scheduling and major airlines are still investing heavily in the area.

Operational Control

Given the wide applications of operations research techniques in airline operation planning and scheduling, it is remarkable that there is relatively little successful applications of operations research in the area of airline operational control. It is only recently that promising work has been done to apply operations research to airline operations control.[4] The result has demonstrated significant improvement in terms of cost saving. This state of affair can be attributed to both the complexity and dynamic characteristic of the problems as we will discuss later on.

This dissertation is another serious endeavor to tackle operational control problem using combined operations research and heuristic methodologies.

1.3 AIRLINE IRREGULAR OPERATIONS

Airline irregular operations, as the name suggests, are just airline operations under irregular situations, which usually means some disturbances or perturbations have happened or are expected to happen in such a manner that they have disrupted or will disrupt the normal or regular operations of the airline. As discussed previously, the sources of perturbations can be many external events. The three most common causes, however, are weather problems, mechanical problems and air traffic control problems (ATCs). Such events occur on a continuous basis and thus airline irregular operations are really a never-ending process.

Since an airline's product is its published flight schedule and it has invested great amount of resources and efforts to build this schedule and other schedules such as crew schedule, maintenance schedule around flight schedule, it is very important to maintain this and other schedules in order to realize the built-in revenues. The objective to the airline during irregular operations therefore is, put simply, to recover the original schedules (flights, crew etc.) as soon as possible so as to minimize the impact to the airline. The disruption to the flight schedule during irregular operations could be in the forms of flight delays, cancellations and aircraft diversions or any combinations of the above. As a result, crew schedule, maintenance schedule could all be disrupted. For instance, flight 1685 takes off at IAH (Houston Intercontinental Airport), arrives at DFW (Dallas/Fort Worth Airport), then flies to LAX (Los Angeles International Airport) and finally comes back to IAH. If, due to mechanical problem, the aircraft can not take off at IAH. Then flight 1685 at IAH has to be either delayed or canceled, causing down-line effects. The schedules of the crew members on the flight are also affected, since they may or may not stay on the same aircraft during the entire trip of this aircraft. It is possible that aircraft maintenance schedule is also affected. This simple example demonstrates the contagious nature of the airline irregular operation problems. It is not difficult to imagine that, during a major perturbation such as when a major hub airport is closed for a significant period of time due to severe weather problems (blizzard, snow storm, for example, which are not uncommon in some areas of the country), the impact to the airline can be disastrous. Such a situation usually involves hundreds of flight cancellations and diversions. It may take the airline several days to fully recover their regular schedules.

During irregular operations, an airline really has two major problems to solve: one is the aircraft balance problem, i.e., change the flight schedule through flight cancellations, diversions and occasionally aircraft ferries so that at a later point in time, the aircraft can be at the right place, at the right time, with the right equipment type to match the flight schedule; another one is the crew balance problem (also called crew pairing repair problem, detailed discussion can be found in next chapter), i.e., to match the crew, in a similar fashion as in aircraft balance problem, to the flight schedule. Since aircraft is the more expensive resource among the two, the current practice in most major airlines during irregular operations is to solve the aircraft problem first (usually done by operations manager with the input from other personnel within the system operations center) and then solve the crew balance problem. Because crew schedule is much more complicated, it is usually the latter that is the bottleneck of the entire problem-solving process.

It should be pointed out that the above description applies mainly to the airlines operating under a hub-and-spoke network system. In airline industry, there have long been two types of airlines. One is the network airline. Most major airlines belong to this category. The other is the point-to-point, linear airline as represented by Southwest Airlines. The advantage of a network airline is that it is a full service airline that can transport passengers between almost any two airports through the hub-and-spoke system using a much smaller number of flights than it would need otherwise (say, in a linear system). The biggest disadvantage of operating a network airline is, however, the enormous complexity involved to serve hubs, to build various schedules and ultimately to fix schedules during irregular operations. The linear airline is much simpler operationally. It can only serve a limited number of airports and is therefore adopted by small airlines. In the case of Southwest Airlines, it operates a single fleet type (Boeing

737), has no hub and uses a very simple schedule. This simplicity in operations means the linear airlines usually have low-cost and can thus offer low-fares.

We now look at the organization of this dissertation. After the overview of airline operations and airline irregular operations in Chapter 1, we describe and discuss in Chapter 2 the crew management process and problem during airline irregular operations, which is the main problem we want to solve in this dissertation. In Chapter 3, we review and compare current different methodologies to solve crew scheduling problem as well as crew problems during irregular operations. The emphasis is on the similarities and differences between the two types of problems which will have profound implications on the design of the algorithms to solve irregular operations crew problems. In Chapter 4, which is the core of this dissertation, we present our model and algorithms to solve irregular operations problem, along with some computational results. Chapter 5 will present a decision support environment that takes advantage of some of the state-of-the-art computing and information technology (IT). As it turns out, this IT environment is critical in the successful implementation and usage of the model developed and it is a challenging work in itself. Finally, in Chapter 6, we discuss some of the works that can be done in the future to improve the work we have done and give some thoughts on the possible direction.

Chapter 2 CREW MANAGEMENT DURING IRREGULAR OPERATIONS

In this chapter, we will describe the crew management process and define the problems airlines face during airline irregular operations. In section 2.1, some terminology that are frequently used in airline industry and in this dissertation are defined and explained. In section 2.2, the crew management problem during irregular operations are identified and defined. Finally, in section 2.3, a comparison is made between the more classical crew scheduling problem and crew control problem for irregular operations. This last section is important in that it helps design the solution methodologies to the latter problem.

2.1 AIRLINE CREW MANAGEMENT AND CREW SCHEDULING

At a major airline in the United States, there are usually several tens of thousands of crew, which include pilots, flight attendants, and ground workers such as machinists, baggage handlers et al. In this dissertation, we are mainly concerned with the first two categories of crew, because during irregular operations, they are the ones that are directly affected and thus are the main concern for crew management during irregular operations. Therefore we always refer to pilots and flight attendants whenever we talk about crew.

Crew are very expensive resources for the major airlines, contributing the second highest component of direct operating cost after fuel. It is therefore extremely important for the airlines to deploy and manage their crew effectively and efficiently.

Crew management, however, is a very complicated task given the large size of the crew at major airlines, their different qualifications and their dispersion at so many different geographical locations. The way the major airlines operate, i.e., hub-and-spoke network system, further complicates the management of the airline operations in general and crew in particular, as will be discussed in the following. Finally, the strict regulatory limitations that Federal Aviation Administration imposes on operating crew as well as union contract, company policies add yet another dimension to the complexity of the crew management.

Crew Bases and Qualifications

Major carriers often have several crew bases in different parts of the country. These include their hub cities and may also have non-hub crew bases in order to deploy their crew in an effective manner. Crew bases also make it easier to manage and train crew. The city where a crew member resides may be different from where he/she chooses his/her crew base. A crewmember's assignment period usually starts from his/her base and end there also.

To cover different markets (routes) and to meet different demands, major carriers usually operate different aircraft types, also called fleet types or equipment types, such as Boeing 737, Boeing 747, DC9 etc. A pilot is usually only qualified to operate on one fleet type, but a flight attendant can generally serve any equipment type. Cockpit crew, or pilots, i.e., captain (CA), first officer (FO) and second officer (SO), can only operate on their qualified positions. Apart from equipment qualifications, there are also other minor qualifications such as language qualification which requires that only crew with particular language ability can serve on certain international route, and airport qualification which dictates that some pilots can not fly certain airports. These limitations have placed strong restrictions on the assignment and reassignment of crew during crew scheduling and operational control.

Hub-and-Spoke System and Crew Management

A hub-and-spoke system affects the crew management in several aspects. First, the flight schedule and thus crew schedule are more complicated in a hub-and-spoke system than in a point-to-point system. This means that the crew schedule is hard to build and hard to fix once broken as during irregular operations. Second, the system and schedule make it almost unavoidable that crew sometimes have to layover at cities away from their domiciles. The airlines also have to provide hotels and transportation for these crew. The length of the time while the crew is on duty may be long, but the actual flight time, the effective duty time, can be short, resulting in low productivity and high cost. Thirdly, the complexities of the schedules and work rules, as will be discussed in the following, force the major airlines to keep a large pool of reserve crew. Reserves, for example, make up 20% of the pilot work force at carriers such as American Airlines.[5] This, again, pushes up airline's already substantial labor cost.

Crew Legalities and Work Rules

By far the most complicated and restrictive factor in airline crew management and scheduling is the federal authority-imposed regulatory rules, union contracts and company work rules. It is not our purpose here to give a complete list of and detailed discussion on these rules. Rather, only an outline and summary of these rules are given. Emphasis is on the impact these rules have on crew management and scheduling. In Appendix, some more important rules are listed as reference. Even there, it is impossible to give all the crew legalities and work rules. In the following, we divide those rules that are most relevant into several categories.

FAR training and certification requirements

Pilot training may include ground training, flight training, flight simulator training, proficiency checks, Cockpit Resource Management, and any other training or qualifying required by Federal Aviation Regulations (FARs), or company policy. There are also airport and route qualification, since some airport or route may be more difficult to fly and thus need special training. Each pilot is usually trained only for one type of equipment and one position. Pilots need training as they move up in seniority to bigger airplanes.

FARs also put minimum flight requirements for pilot in order to stay certified. For instance, a pilot must have at least three takeoffs and landings in 90 days.

FAR flight crew duty period time and flight time limitations

FAR also requires strict duty time and flight time limitations for safety. Duty period time refers to the elapsed time from the time a flight crew member is required to report for duty (or deadheading to or from duty) or the actual reporting time, whichever is later, until the time the pilot is released from duty from the last flight segment flown or deadheaded before a minimum rest period or a day off (see below). Flight time is the time the first movement of an aircraft for the purpose of flight until it comes to rest at the next point of landing. The FARs on duty period time and flight time limitation are very detailed and are different for domestic and international flights. The main points are summarized below.

Duty period time: For both pilot and flight attendant, the scheduled duty period time in domestic flights can not exceed 14 hours during regular operation and can not exceed 16 hours during irregular operation. (At the time of this writing, FAA is mulling to impose tougher restrictions, reducing the maximum duty time to 14 hours instead of 16.)

Flight time: For pilots only, the maximum accumulated flight time can not exceed:

- 8 hours between required rest periods
- 30 hours in any 7 consecutive days
- 100 hours in any calendar month
- 1000 hours in any calendar year

It should be pointed out that the above flight time limitations refer to the scheduled and regular operations. In practice, it works in this way, take 30 hours in 7 days limitation, on the seventh day, if the irregular operations happens and according to original schedule, the accumulated flight hours in the 7-day period will exceed 30 hours, it is acceptable; if, however, the limitation would be exceeded due to rescheduling, it is not acceptable. This is called in the airline industry six-day look-back policy. Other limitations work in a similar way.

For international flights, since it usually takes longer, the limitations for both duty period time and flight time are different from the domestic ones. Some of the limitations are given in Appendix.

FAR pilot minimum rest time requirements

In connection to the maximum duty and flight time limitations, FARs also stipulate strict crew-rest requirements for pilots in any 24-hour period. The following table summarizes the FAR minimum rest requirements. Again, these are just for domestic flights and international flights have different requirements.

The flight time in 24 hours is compared to the above table to determine the rest required during that 24 hours. A normal rest may be reduced if another rest (mandatory) starts within 24 hours after the start of the reduced rest.

Table 2.1 FARs domestic minimum rest requirements for pilots

Scheduled Flight Time	Normal Rest	Reduced Rest	Mandatory Rest
Less than 8:00	9:00	8:00	10:00
Less than 9:00	10:00	8:00	11:00
Equal or greater than 9:00	11:00	9:00	12:00

Flight time is computed block to block using scheduled times. For each flight leg, flight time is accumulated from the arrival time to the 24 hours preceding the arrival. Block to block is defined to be the period of time beginning when an aircraft first moves from the blocks for the purpose of flight, and ending when the aircraft comes to a stop at the blocks at the next point of landing, or at the point of departure if the flight returns without becoming airborne. Rest time is computed brief to debrief using actual times (if available) or scheduled times.

For example, with a total of 7 hours of flight time in 24 hours, a pilot would normally receive at least 9 hours rest during that 24 hours. The rest period may be reduced to not less than 8 hours. When the rest is reduced, the pilot must have a mandatory rest of not less than 10 hours within the next 24 hours after the start of the 8 hours reduced rest.

The rest time requirements for flight attendant are much simpler. They are 10 hours block in to block out at their crew base and 8 hours and 45 minutes block in and block out at a layover city.

Off time and vacation policies

Off time and vacation policy are determined between airlines and their crew, usually written into their union or labor contracts. Off time can be movable and immovable. In the first case, if a crew member's duty time extends into his/her off time, the off time can be shifted accordingly or moved to a different time slot. In the second case, however, the off time is guaranteed and can not be moved. Thus, if an irregular operation happens and the duty time would overlap with the immovable off time after reschedule, then this reschedule will not be acceptable.

In many airlines, if a pilot's duty period overlaps with a scheduled vacation by even one day due to irregular operation, then the vacation is canceled with pay. If, say, the scheduled vacation is 15 days, it means there is a possibility that the pilot will get paid for 14 days without flying and, the company will have to schedule another vacation for the pilot This is an extreme case, of course, but it does point to the inherent cost caused by the strict and sometimes inflexible work rules, which is a demonstration of the power of union (especially pilot union) in airline industry and the legacy of the regulated era when efficiency and competition were not big concerns.

Seniority rules

In airline industry, seniority of crew is another important issue. Seniority determines crew's pay rate, their priority in selecting "duty trip" (duty assignment), and their choice of off time and vacations. Airlines are bound by labor contract to assign duty period and crew "bid" their scheduled duty period according to seniority.

Crew Scheduling

Given the flight schedule, crew scheduling is the mapping of all flights in the schedule into a set of crew trips such that each flight segment (or leg) is covered at least once, and optimally only once. A mapping consists of a number of pairings, which are sequences of flight legs that begin at a crew base station, fly around the system, and return to the original base station. A crew pairing can last from one day to over ten days, but three or four days are most common. A crew schedule is usually built on a fleet-by-fleet basis because pilots are qualified to fly only one type of aircraft.

The challenge of crew scheduling is to build a crew schedule that utilizes the crew resource most efficiently and, at the same time, conform to the very restrictive and complex crew legalities discussed above. At most major airlines, union contracts specify that flight crew will be guaranteed pay for some number of hours each day or each trip. Airlines must try to build their crew schedule in such a way that each and every crew's schedule meet or exceed his/her pay guarantees to the maximum extent possible. Obviously, a trip lasting several days that contains very short flying time is very expensive for the airline, because the crew will receive pay over and above the actual flying time.

At most airlines, crew schedule is done on a monthly basis, in correspondence to flight schedule. Once airlines decide their monthly flight schedule, they build the crew schedule and publish all the pairings that are built based on the flight schedule. These pairings collectively cover all the flight in the flight schedule. Most airlines then group these pairings into packages containing one month of flight assignments, also called "bidlines". Each individual crew can then "bid" his/her monthly schedule by selecting their choice of duty time according to their seniority. Pairings that are not awarded or assigned, as well as those that an assigned crew member cannot fly for any reason, are flown by reserve crew.

At present, crew scheduling is done at major airlines using sophisticated operations research models and powerful computers. Despite of these, achieving optimal crew schedule has largely remained a dream because of the complexity of the problem. (We will discuss methodology of solving crew scheduling problem in next chapter.) As such, there is still many room for improvement in crew scheduling.

2.2 CREW MANAGEMENT DURING IRREGULAR OPERATIONS

As has been pointed out previously, the flight schedule of an airline is its sole product. All the airlines' operational activities are centered around this goal—delivering this product to the passengers successfully and smoothly, and all resources—aircraft, crew, gate—have been aligned and coordinated towards this goal. As has been discussed, building flight schedule and the corresponding crew schedule, gate plan are a complicated and time-consuming process at major airlines. Any disruption to the execution of these schedules represents a loss of the airlines' revenue and reputation. The single most important goal of an airline during irregular operations, therefore, is to recover its various schedules as soon as possible. To achieve this goal, the airline controllers need to reschedule and reassign the two most important resources: aircraft and crew that are displaced/disrupted spatially and/or temporally due to irregular operation in such a way so that after some time these resources again are back on "schedule".

At present, the practice at major airlines is to organize the irregular operation activities along the lines of fleet types and functions. Each fleet type is assigned an operations manager who takes overall charge of the fleet type. The operations manager also pairs with a crew coordinator who is responsible for monitoring and rescheduling of the affected crew in that particular fleet.

During irregular operations, operations manager usually comes up with a recovery plan for the flight schedule and aircraft, which is the more expensive and scarcer resources, and then the crew coordinator helps the operations manager make the decisions by finding out a crew recovery plan. This simplified description of the process, however, captures two important characteristics of irregular operations crew management currently practiced at major airlines. First, aircraft is the more important consideration in any recovery plan; Second, crew coordinator basically reacts to the decision made by operations manager.

Operations Manager and Irregular Operations

Let us first look at the types of decisions the operations manager makes during irregular operation regarding flight schedule and aircraft.

The most frequent problems causing irregular operations are the following:

- Mechanical problems
- Weather problems
- Air traffic controls (ATCs).

Mechanical problems happen all the time. ATC problems are seasonal. Weather problems are worst in winters when blizzard or snow can cause the closure of airports. Summers have thunderstorms.

When a problem arises, the operations manager usually has the following options:

- Flight delays
- Flight cancellations
- Equipment substitutions
- Ferry-in's
- Aircraft diversions

When considering different alternatives, delay is first considered. When choosing which flight(s) to delay and by how long, operations manager needs to consider many factors and protects the more important markets (routes). Priority can be decided by the following factors, for example:

- International markets—these flights have the highest priority.
- New market—because of the marketing consideration, new market should have higher priority.
- Shuttle market—in some high frequency flight market, e.g., between New York and Boston, it is relatively easy to simply cancel a flight instead of delaying it.
- Yield—based on historical data, markets with higher yield should be protected.

When something happens, the operations manager usually waits some time (called "info-time") for the problem to be identified. After the cause of the problem is identified, the operations manager then weighs his various options. When making the "delay or cancel" decisions, he will also consider, besides the priorities given to some markets, the following criteria, e.g.:

- Customers (Are customers still there?)
- Crew legalities (Can the crew be rescheduled?)
- Down-line impact (How many flights will be impacted?)

To handle the aircraft problems caused by delays, the operations manager can either wait until the problems are automatically solved by absorbing the delays over time or he/she could do some equipment substitutions. The following example illustrates the use of equipment substitution to solve a delay problem.

The portion of flight schedule given in Table 2.2 represents two aircraft, with tail numbers of 146 and 370 respectively. Flight 923 departs from Dayton (DAY) at 10:39 AM local time and arrives at Chicago (ORD) at 10:40 AM after 61 minutes. The same aircraft continues as flight 923 and leaves Chicago 35 minutes later at 11:15 AM for Steamboat Springs (HDN). Another aircraft with tail number 370 also leaves Chicago at 12:10 PM. If flight 923 is delayed for about one hour from DAY to ORD, then aircraft 146 will miss its next flight from ORD to HDN departing at 11:15 AM. The operations manager can now either delay flight 923 at ORD, and possibly more flights down-line, or he can substitute aircraft 370, which is already at ORD and waiting for the departure of flight 418 at 12:10 PM, for aircraft 146. When aircraft 146 arrives at ORD at around 11:40 AM, it will substitute aircraft 370 and continue as flight 418. This equipment substitution has the advantage over the multiple delays down-line in that only one flight, i.e., flight 923 from DAY is delayed and all other flights leave on schedule.

Table 2.2 Portion of a flight schedule

Leg	**Flt No**	**Date**	**Dpt City**	**Dpt Time**	**Arv City**	**Arv Time**	**Flt Time**	**Equip Type**	**Tail No**
1	923	960214	DAY	1039	ORD	1040	61	727	146
2	923	960214	ORD	1115	HDN	1309	134	727	146
1	418	960214	ORD	1210	CLT	1452	102	727	370

Cancellation is the simplest decision to make but may result in loss of revenue and customer goodwill. Once a cancellation decision is made, the next step is to recover the system as soon as possible. The way to do it is to simply cancel a corresponding flight (e.g., if a flight from Newark to Boston is canceled, the a flight from Boston to Newark can be canceled to balance the system) or to ferry in another aircraft from other stations or use the spare aircraft. In practice, ferry-in's happen quite frequently.

When some severe disruption happens, such as when a major hub is closed for one or two days, there will be mass delays, mass cancellations and diversions in the system (diversion means a flight arrives at an airport different from its original destination, possibly due to closure of the airport). This kind of situation takes operations manager significantly more time to recover. The practice is usually to set a target time—often the beginning of a daily schedule—by which time the system will be recovered. Before that target time, the operations manager, together with crew coordinator and other personnel at system control center, will try to work out a recovery plan. Often time, the aircraft problem is relatively easier to solve, but the crew problem is much more difficult to solve.

Crew Coordinator and Irregular Operations

Once the operations manager proposes a recovery plan after considering all the factors discussed above. It is the crew coordinator's responsibility to make the necessary changes in crew assignments to accommodate the schedule changes. This process is usually called at airlines "crew pairing repair", because the goal is to find solutions to fix the broken crew pairings due to flight delays, cancellations, and diversions. This is often a very time-consuming process and takes anywhere from a few minutes to a few hours, depending on the severity of the disruption. Sometimes, there may not be a feasible solution at all given the number of crew members involved and the constraints of the complicated work rules or crew legalities that must be checked for the crew members concerned. In this case, the operations manager will have to modify his/her recovery plan. Thus this is an interactive process and can take many iterations before the operations manager can come up with a viable recovery plan.

Table 2.3 Segment of crew pairing for crew A before the crew swap

Seq No	**Date**	**Flt No**	**DH**	**Equip**	**Org**	**Dst**	**Dpt/ Bref**	**Arr/ Derf**	**Flt Time**	**Duty**	**F/S**	**Tailnum/ Layover**
120	29	1623		DC9	EWR	BWI	1635	1741	0106		CL	544
130	29	1623		DC9	BWI	GSO	1815	1921	0106			544
140	29	1623		DC9	GSO	ATL	1945	2055	0110			544
150	29	1623		DC9	ATL	IAH	2115	2220	0205			544
160		DP03					1605	2235	0527	0730		

For a crew coordinator, his/her options in repairing disrupted pairing includes crew swap, deadheading, using reserves crew, sit crew or layover crew.

Crew swap refers to the situation when two crew members (or groups) swap their assignment for a few flight segments and then go back to their original schedules again. For the swap to be valid, these two crew members must be able to fulfill their changed schedules in accordance with any binding legalities. Deadheading refers to the situation where a crew member has to take a non-serving flight to his/her operational assignment or base. The following example illustrates a case when both crew swap and deadheading happen.

Table 2.4 Segment of crew pairing for crew B before the crew swap

Seq No	**Date**	**Flt No**	**DH**	**Equip**	**Org**	**Dst**	**Dpt/ Bref**	**Arr/ Derf**	**Flt Time**	**Duty**	**F/S**	**Tailnum/ Layover**
010	29	1722		DC9	GSO	BWI	1342	*1442	0100			544
020	29	1722		DC9	BWI	EWR	1445	1545	0100		CL	544
030	29	1697		DC9	EWR	ORF	1815	1941	0126			563
040	29	1697		DC9	ORF	CHS	2015	2130	0115			563
050	S1	DP01					1220	2145	0440	0925		1315
060	30	1544		DC9	CHS	ORF	1130	1235	0105			537
070	30	1544		DC9	ORF	EWR	1300	1418	0118			537
080	30	1651		DC9	EWR	ORF	1500	1618	0118			541
090	S1	DP02					1100	1633	0341	0533		1322

Table 2.3 and 2.4 show portions of pairings for crew A and B. The current time is 13:07 local time at Houston (IAH) on the 29th of the month. Crew A is based at Houston and is into his third and last duty period. He is scheduled to return to IAH at the end of the duty day. Crew B, on the other hand, has just started his pairing from his base Greensboro (GSO) and has completed his first flight leg from Greensboro to Baltimore (BWI), indicated by an asterisk before the arrival time. Now the operations manager for the fleet type DC9 has decided to cancel flight 1722 from Baltimore to Newark (EWR) leaving at 14:45 PM due (possibly) to mechanical problem. This is indicated in the Flight Status column of the pairing table by CL. To balance the aircraft flow, the operations manager has also decided to cancel flight 1623 back from Newark to Baltimore departing at 16:35 PM (this is called cancel-in/cancel-out by airline folks) which uses the same aircraft (tail number 544) as flight 1722. These cancellations disrupted the

pairings for crew A and crew B, who could not get to Newark and Baltimore, respectively, to continue their trips. The trick here is to switch some of their flight segments and then get back to their original pairings at a future time, possibly using deadheading. In this example, crew A will take the next four flight legs of crew B's pairing after flight 1722 and ends up at Newark on the 30th of the month. To return his base at Houston, he will take a deadheading flight from Newark to Houston. Crew B, meanwhile, will take the next three legs of crew A's pairing after flight 1623. He will end up at Houston in the evening of 29th, though. To continue his pairing next day at Newark, he will have to deadhead on a flight from Houston to Newark next morning. Tables 2.5 and 2.6 give the new, repaired pairings for crew A and B respectively.

Table 2.5 Segment of crew pairing for crew A after the crew swap

Seq No	**Date**	**Flt No**	**DH**	**Equip**	**Org**	**Dst**	**Dpt/ Bref**	**Arr/ Derf**	**Flt Time**	**Duty**	**F/S**	**Tailnum/ Layover**
120	29	1623		DC9	EWR	BWI	1635	1741	0106		CL	544
130	29	1697		DC9	EWR	ORF	1815	1941	0126			563
140	29	1697		DC9	ORF	CHS	2015	2130	0115			563
150		DP01										
160	30	1544		DC9	CHS	ORF	1130	1235	0105			537
170	30	1544		DC9	ORF	EWR	1300	1418	0118			537
180	30	205	D		EWR	IAH	1530	1808				
190		DP03										

Note that the symbol 'D' in the DH (DeadHeading) column indicates that flight leg is a deadheading leg. The equipment columns are left blank intentionally, because they can be any equipment types. Also the corresponding accumulated flight time, duty time as well as rest/layover time will have to be recalculated. Finally, in this crew swap, crew A's pairing has been extended by one day. This may not always be feasible to do.

Table 2.6 Segment of crew pairing for crew B after the crew swap

Seq No	**Date**	**Flt No**	**DH**	**Equip**	**Org**	**Dst**	**Dpt/ Bref**	**Arr/ Derf**	**Flt Time**	**Duty**	**F/S**	**Tailnum/ Layover**
010	29	1722		DC9	GSO	BWI	1342	*1442	0100			544
020	29	1722		DC9	BWI	EWR	1445	1545	0100		CL	544
030	29	1623		DC9	BWI	GSO	1815	1921	0106			544
040	29	1623		DC9	GSO	ATL	1945	2055	0110			544
050	29	1623		DC9	ATL	IAH	2115	2220	0205			544
060		DP01					1220					
070	30	132	D		IAH	EWR	0940	1402				
080	30	1651		DC9	EWR	ORF	1500	1618	0118			541
090		DP02										

The above example demonstrates the simplest crew swap, i.e., two-way swap. There are times when more complicated swaps, such as three-way swaps or even swaps involving more crew can be used.

Reserve crew is another important asset in the crew coordinator's tool-set. But use of reserve crew should be careful. First, using reserves costs money, since even though reserve crew has been paid for their availability, they are not paid for the rate if they are actually flying, which will be applied when they are used. Secondly, reserve crew are not always available at every station. The airlines usually only keep a reasonable number of reserve crew at crew bases. Thirdly, the airlines sometimes will not allow the use of reserve crew arbitrarily in many cases in anticipation of a major disruption when large number of reserves are needed to recover the system.

2.3 CREW SCHEDULING PROBLEM VS. CREW CONTROL PROBLEM

Based on our discussions on crew scheduling and crew management problems during irregular operations in previous two sections, we can make some comparison between the two kinds of problems in the hope that we can gain some insight on the nature and characteristics of the problems. This will have implication to our solution methodology for the crew control problem in irregular operations. The attempt to apply the solution approaches of crew scheduling problem is natural and strong. After all, the problem facing crew coordinator during irregular operations is to reschedule the crew whose pairings are disrupted. Table 2.7 compares the them.

The biggest difference, and also the strong connection between the two problems, is that crew scheduling problem is to build a schedule from scratch while crew control problem is to fix a schedule that has already been built. Crew scheduling needs to build the entire schedule while crew control just needs to focus on where problems arise; the former has to schedule an entire month while the latter needs only to deal with a much shorter time horizon. The biggest challenge to crew control during irregular operation is to come up with solutions quickly—late solution is meaningless. Very often, the crew coordinator will need to recommend to operations manager different alternatives. As a result, crew coordinator can not afford to spend a lot of time finding optimal or even a better solution. A feasible solution will usually be good enough. Spending more time in the hope of getting a better solution is almost certainly seeking diminishing return.

Table 2.7 Comparison of crew scheduling and crew control problems

	Scope	**Time horizon**	**Solution time requirement**	**Solution quality requirement**	**Solution quantity requirement**
Crew Scheduling Problem	global, entire system, all crew, all airports	long, usually months	not restrictive. usually a few weeks	high. optimal or close to optimal preferred	one
Crew Control Problem	local, may involve only a few airports	short, from a few hours to a few days	restrictive. should not take more than a few minutes	reasonable. good feasible solutions are acceptable	multiple

Chapter 3 Current Solution Methodologies for Crew Management Problems

In Chapter 2, we discussed the crew management process and problems at major airlines. We basically categorize the crew management problems into two classes: the crew scheduling problem and the crew control problem during irregular operations. Throughout this dissertation, we also refer to the crew scheduling problem and the crew control problem as crew pairing optimization problem and crew pairing repair problem respectively when the emphasis is on solving the problem by various algorithms. In this Chapter, we will give an overview of the current solution methodologies for these two types of problems and will also compare the different approaches in solving these problems.

3.1 SOLVING CREW SCHEDULING PROBLEM

Crew scheduling problem at major airlines is mostly solved by complicated integer linear programming(ILP). It is also called crew-pairing optimization. A crew pairing is a sequence of flight legs that starts and ends at a crew base and typically lasts from two or three days. A crew member works four or five pairings per month. The goal of crew scheduling is to build a monthly schedule that makes up pairings which cover all flights and minimize the total cost. The pairings must conform to FARs, company policies and labor contracts. The crew scheduling problem is modeled as a set partitioning problem, where the rows represent flights to be covered and the columns represent candidate crew pairings. While the crew legalities decide whether a particular pairing is valid or not, the cost of the pairing affects its desirability.

The crew-pairing optimization problem can be formulated as the following:

$$\begin{aligned} \text{Minimize} \quad & c^{p} x \\ \text{Subject to} \quad & Ax = 1 \\ & x = (1,0) \end{aligned}$$

Each row in the matrix A represents a flight segment and each column represents a pairing. And

$a_{ij} = 1$ if segment i is covered by pairing j, $a_{ij} = 0$ otherwise;

$x_j = 1$ if pairing j is part of a solution, $x_j = 0$ otherwise;

$c_j =$ the cost of pairing j.

In the above problem formulation, we essentially requires that each flight segment be covered once and only once. Thus it is a set-partitioning problem. We may also allow crew deadheading, i.e., transporting crew as passengers. In this case, the problem will be formulated as a set-covering problem, changing the constraint $Ax = 1$ to $Ax \geq 1$. Both set-partitioning and set-covering problems are NP-hard combinatorial problems.[6] At major airlines, there are usually tens of thousands of flight crews serving thousands of flights a day. It is impossible to solve set-partitioning or set-covering problem with such a size. The current practice is to break the problem into sub-problems with smaller number of pairings and flight segments.[2] Solving the integer linear programming problem, however, is still the relatively easier part. The more difficult part, as is always the case in any real-world problems, is to incorporate all the complicated crew legalities and work rules into the solution. Before solving the integer programming problem, a set of valid candidate pairings must be generated to feed into the set-partitioning model. Finally, various cost factors must be taken into consideration in order to select the more desirable pairings and minimize the total solution cost. We discuss them separately in the following.

Sub-problem Selection

Because even for the smallest fleet, it is impossible to obtain a globally optimal solution due to the large size and the combinatorial nature of the problem. It is therefore necessary to solve a series of sub-problems with smaller number of flight segments and pairings so that a reasonably-sized A matrix can be built and the sub-problem can be solved to optimal. In general, the larger the sub-problems, the better the final solution will be. To select sub-problems, an initial solution is first generated. This initial solution can be produced manually, using some heuristics or simply adapted from previous month's schedule. It does not need to be of high quality or even legal, it merely serves as a starting point. The next step is to select, either randomly or systematically, a few number of pairings from this initial solution to form a sub-problem. All other pairings in the initial solution are locked out of the sub-problem. From the flight segments within the sub-problem, all possible pairings (columns) are exhaustively generated. The set-partitioning model is then applied at this stage to find a set of newly-generated pairings that gives the minimum cost for the sub-problem. This new set of pairings will replace the originally-selected one in the initial complete solution. These pairings, together with pairings that were locked out of the current sub-problem, form the new initial solution. The process is then repeated, producing better and better feasible complete solutions, until some stopping criterion is met.

The problem with sub-problem selection is, even if the optimal solutions for each sub-problem are found, the global optimal solution is still not guaranteed since the size of sub-problems are limited to about 100 segments and 10,000 columns [7]. This is a far cry from the tens of thousands of segments and trillions of columns that are needed in a global problem. On the other hand, this implies that there is still much room in improving the quality of the crew schedule and thus further reducing crew cost at major airlines.

Pairing Generation

While the solution algorithm is critically important in solving crew scheduling problem, the most crucial step in building a good crew schedule is the column or pairing generation. This is both because of the enormous number of possible pairings that can be generated and the complex crew legalities and work rules that must be considered in building pairings. Without a mechanism that can generate good quality pairings, it is meaningless for the subsequent optimization. While the optimization algorithm is standard and mechanical, pairing generation needs experience and heuristics.

All the pairing generated must conform with FARs, company policies and labor agreements as discussed in Chapter 2.

Cost

The selection of desirable pairings, besides crew legalities and company work rules, is ultimately determined by the cost of pairings. The major cost components associated with crew are the so-called pay and credit, hotel and per-diem expenses. It is also necessary sometimes to introduce artificial cost into the model so as to price out some solutions with unfavorable characteristics and to assign lower cost to some preferable solutions.

Pay and credit

This is by far the largest cost component. It is the guaranteed hours of pay minus the actual hours actually flown. These hours are nonproductive crew time and are therefore to be minimized. At most airlines, company policy or labor contracts guarantee that crew must be scheduled a minimum flying time per duty period or a guaranteed average flying time per day. Such guarantees were negotiated to discourage management to schedule a very short duty day. Thus even though the crew fly less than the guarantees, they are paid as though they did. Also, most labor contracts may also guarantee a certain minimum percentage of flying time in a duty period and/or a minimum percentage of flying time in the entire pairing to discourage long sit time (time between two flights within a duty period) and long layovers. Usually deadheading is also discouraged since the crew is paid 50 percent or more for the flying time of the flight segment even though those are unproductive times for crew. A good pairing is therefore the one that meets these minimum guarantees, reduces the unproductive crew time and thus the cost. In practice, it is difficult to avoid entirely pairings with deadheading, short duty days, long sit time or long layover because flight schedule is mostly market-driven and equipment utilization has more leverage in terms of cost reduction than crew schedule.

Per-diem and hotel expenses

If crew is scheduled to stay over night at stations other than their home bases, the airline must provide a hotel room for each member of the crew, and if the hotel does not provide a courtesy van to and from airport, it must also provide a limousine or taxi as well.

Per diem expenses are cost associated with crew members for meals while away from base and are considered legitimate business expenses.

The above solution methodology is currently the most successful one and indeed, is the one adopted by the popular TRIP (trip reevaluation and improvement program) crew-pairing optimization system which was first developed by American Airlines and IBM in the early 70s'. TRIP has been used by a dozen major airlines and railroads.[3]

There are also other solution methodologies used to solve crew scheduling problem. In recent years, constraint logic programming (CLP) is another methodology that has been applied to solving crew scheduling problem and has been reported with positive results. Because of its large potential for cost saving, major airlines will continue to invest substantially in terms of money and manpower in the improvement of solving crew scheduling problem.

3.2 SOLVING CREW CONTROL PROBLEM

Compared with the sophisticated solution methodologies used for solving crew scheduling problem, solving crew control problem during irregular operations has long been a realm in which crew coordinators' experience and intuition are the most important aid. The whole process is still manual and paper-based. At major airlines, there is not any sophisticated decision support tool to help a crew coordinator with crew control. This is largely determined by the nature of the operational control problem.

As has been discussed in Chapter 2, the problem of controlling an airline's operation is inherently reactive by nature. It is very unpredictable and almost anything can go wrong during operation. For crew coordinator, anything that affects the schedule will ultimately affect crew. Almost any change of flight schedule as a result of irregular operations requires change of crew schedule. The crew coordinator thus needs to repair the broken pairings caused by the broken flight schedule, find crew for all the flights that are disrupted and make sure the solutions do not violate any of the crew legalities or work rules. All these have to be done within a very short time period so that the airline operation can quickly get back on its original schedule. Often, coming out with a feasible solution is challenging enough and rarely the crew coordinator has the luxury to find better solutions, let alone optimal ones.

The first step in supporting a crew coordinator's work is to provide various real-time information, such as flight data, crew data, airport information, hotel information. These data and information should be organized and displayed in very convenient formats (in graphical formats, for instance) for quick access. For example, if flight 1649 at Chicago is canceled, and the crew on it is to arrive at Detroit to serve flight 495, the crew coordinator need to quickly find out if the crew can be sent to Detroit on another flight from Chicago in time for the crew to catch flight 495 or; if it is not possible, are there reserve crew or layover crew at Detroit who can substitute the original crew? The crew coordinator can also display all the incoming flights at Detroit and search for a swap solution. Another way to help the crew coordinator is to add monitoring and alerting functions to the system. This will give crew coordinator advance time to react to possible disruption. For example, if an upstream flight is severely delayed and it may become illegal for the crew to continue to serve the last two flight segments of the original pairing. The crew coordinator is then alerted and can take precautionary measures (finding reserve crew for the affected flight legs, for example) to deal with the situation.

All these real-time information display, monitoring and alerting functions require a modern, sophisticated information technology and communication infrastructure.

Up to now, there has been a lack of sophisticated, model-based decision support tool for crew control. Obviously it is impractical to apply the solution methodologies for crew scheduling problem directly to crew control problem due to the requirement of quick response time during irregular operations. Even though these two classes of problem bear some similarities, there are still significant differences between the two. Often, crew control requires the capability of what-if analysis for evaluating different alternatives, it also needs multiple solutions. In the event of no feasible solutions, it will accept partial solutions since the airline operation can not be stopped and solving part of the problem is better than no solution at all. In the crew scheduling problem, however, it is usually an all-or-nothing approach.

In recent years, there has been some efforts in the application of artificial intelligence techniques such as constraint logic programming to solve irregular crew management problem in airline operation. These efforts have been reported to lead to some success.[8] But they seem to solve relatively small problems in terms of number of flight segments and number of crew members involved and, short time horizon, e.g. daily operations. This implies that at least in the following two scenarios these models will not be able to solve the problem or extremely stretched: 1) in a major perturbation of the system, such as when a major hub is closed, the problem quickly propagates to the entire system and; 2) the short time horizon means that the solution is relatively short-sighted such that it may cause severe problem later on. But it is exactly in these two scenarios that crew coordinator has difficulty in solving the problems quickly, which then translates into huge cost for the airline.

There are a number of authors who have discussed airline irregular operations and developed models or framework for various aspects of the irregular operations. Luo and Yu [9] studied the airline schedule perturbation problem caused by the FAA's ground delay program. They provided models and several algorithms for take-off/landing slot re-assignment. Yu [10] and Argüello et al. [15]

have developed a system-wide model for aircraft re-routing during irregular operations. Their models put both flight delays and cancellations into a unified framework. Their results are generally very encouraging. Mathaisel [11] proposed an integrated decision support framework and discussed some system features.

In the next chapter, we will formally introduce our model and algorithms for the crew control problem during airline irregular operations. We believe our effort is the first serious one to solve large-scale crew control problem and, as will be seen in next chapter, our approach will address some of the deficiencies by previous work.

Chapter 4 Model and Algorithms for Solving Irregular Operation Crew Problem

In this chapter, a model is proposed for crew management during airline irregular operations. Several algorithms for solving the problem are also discussed. Finally, computational results based on the algorithms are presented.

4.1 THE MODEL

As have been discussed in previous chapters, the goal of crew management during airline irregular operations can be summarized as to recover the entire system, at minimum cost and with minimum disruption to the system, as soon as possible. By recovering the entire system, we mean that after a "recovering period", the airline should resume their painfully built original schedules, including flight schedule and crew schedule. For crew, each crew member should be where he or she should be according to his or her schedule after the system is recovered and from then on, crew will continue their original pairings. During the recovering period and before the system is entirely recovered, some crew's pairings may be modified to accommodate the flight schedule changes due to irregular operations. The disruption to the crew pairings should be minimized, since crews are humans, they will complain if their schedules are disrupted too much and, the airline also has an interest in seeing that the original schedule is not perturbed too much. For example, if a crew member is scheduled to layover at a certain city and a hotel room has been reserved, now if this crew is being rescheduled to layover at another city, the airline will have to arrange another hotel room for the crew—if there is room at all. The cost consideration is also important in rescheduling crew. All the cost factors in crew scheduling discussed in previous chapter still apply here. If there are alternative recovery plans, a lower cost solution will definitely be preferred—if it is possible to search for a lower cost solution given the time constraint.

The model that is proposed here is an airport-time network model as shown in Figure 4.1 on the next page. It represents a snapshot of the entire system within a given time window. The start of the window can be the current time and the end of the window is the proposed time by which the entire system has been recovered. The components of the network are explained below.

Nodes: At each airport, there are four different types of nodes.

Crew Nodes: The crew nodes represent either crew who is originated at the airport when the problem starts or arrival crew. The original crew is placed at the time when they are available and the arrival crew at the time of their arrival. The original crews are indicated by triangles before crew nodes.

Flight Nodes: The flight nodes represent the departure flights, and they are placed at the second column of each airport and at the scheduled time of departure.

Reserve Nodes: The reserve nodes represent the availability of reserved crew, and they are placed at beginning of the time window when they are available to serve.

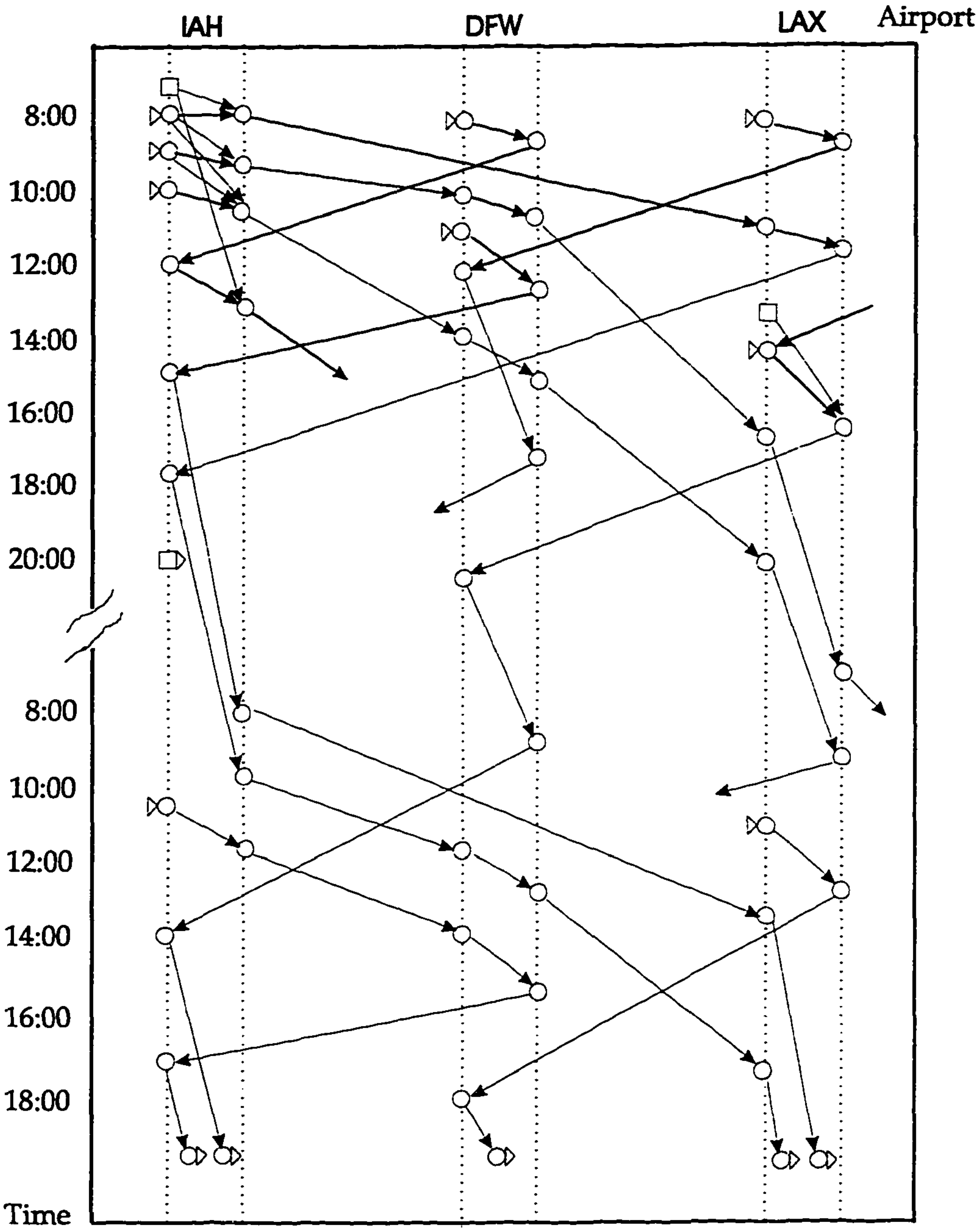

Figure 4.1 Airport-time network for crew management model during irregular operations.

Return Nodes: The returning nodes are used to force the crews to return to their original schedule after the recovery time. For example, if the targeted recovery time of the system is 8:00 PM the next day and a crew member is scheduled to be at Houston (IAH) by that time, then a return node for the crew member is placed at IAH at 8:00 PM the next day. Only original crews have return nodes. Return nodes are indicated by a triangle to the right of the nodes. It should be noted that the number of return nodes is equal to that of the original crew nodes.

Arcs: Arcs are used to connect various pairs of nodes. There are five different types of arcs in the network.

Scheduled Arcs: These arcs emanate from crew nodes to their originally scheduled flight nodes. These arcs represent the original schedule.

Swap Arcs: These arcs emanate from crew nodes to flight nodes that are not their originally assigned flights. Obviously, only when the flight node is later in time than the crew node can there be a swap arc between them. A parameter may also be defined such that only flight nodes that are within a certain range of time from the crew node can have a swap arc from that crew node.

Flight Arc: These arcs represent the flight from one airport to another. They originate from flight nodes at departure airports and end at the corresponding crew nodes at their destination airports.

Reserve Arc: These arcs emanate from reserve nodes to those flight nodes at the same airport which can be served by the reserve crew.

Return Arcs: These arcs represent the returning of crews to their corresponding return nodes. They emanate from crew nodes at the airport where their corresponding return nodes are placed to the their return nodes. Of course, only crew nodes earlier in time than their corresponding return nodes can connect an arc to the return node.

There is some flexibility in the model to accommodate different situations. The costs of the arcs can be assigned to reflect either the preferences or penalties of certain arcs. Also, apart from the limit of a time range to assign swap arcs, it is sometimes useful to limit the swap arcs to the same fleet type when the crew nodes represent pilots who can only operate one equipment type.

We now look at how to solve the problem based on the model given in previous section. We explore two different approaches and compare their distinctive advantages and disadvantages. The first approach is the traditional OR techniques and the second one uses heuristics. Before we start our discussion on algorithms, we define the model to be a single equipment model in that we only solve problems within one equipment type, since pilots are qualified for only one type of equipment and the crew schedule is also built by fleet type. This means the crew nodes in the model all belong to a given equipment type, but the flight nodes are for all the flights since crew can deadhead on a flight of different equipment type.

4.2 OR APPROACH

Due to their obvious similarities, it is very natural to adapt the algorithm used in crew scheduling to solve the current problem. The problem can be formulated as follows:

$$\text{Minimize} \quad c^p x$$

$$\text{Subject to} \quad Ax = 1$$

$$\sum_{j_i} x_{ij_i} = 1 \qquad i = 1 \text{ to } m$$

$$x = (0,1)$$

where i is the index for ith crew member, m is the total number of original crew. j_i is the index for all possible, different legal paths for crew i. These paths start from the crew node for the original crew i and end at the return node for crew i.

Compared with the set-partitioning problem for crew scheduling, there is an additional type of constraints in the current formulation: the crew flow conservation constraints. These constraints force each crew member to reach their return nodes. These constraints further complicate the set-partitoning problem, which is itself an NP-complete problem. The difference here is that instead of different pairings in the crew scheduling problem, we have different clusters of pairings. Each cluster of pairings corresponds to all the possible legal pairings for a particular crew member and one and only one pairing should appear in a solution.

Not surprisingly, we can basically follow a similar approach to the one discussed in previous chapter on crew scheduling solution method. Recall that two important steps in solving crew scheduling problem is subproblem identification and pairing generation. These steps still apply to the crew model for irregular operation. They are discussed separately in the following.

Sub-problem identification

In the crew scheduling problem, the sub-problem approach is basically by virtue of necessity: it is simply impossible to solve such a large problem for the entire system that spans a month. By dividing the whole problem into a series of manageable sub-problems, the problem becomes tractable, although in so doing it is difficult, if possible, to obtain globally optimal solution. In the crew control problem, most problems occur locally, i.e. most problems concern only a few airports or flights. Indeed, it is difficult to image why one would need to include airports and flights in the eastern part of the country if severe weather has caused some flights either canceled or delayed at Austin, Texas. Also, the duration of the problem or the recovery time window should be significantly shorter than a month, from a few hours to a few days. As a result, this "local vs. global" and "short-term vs. long-term" comparison obviously benefits the solving of crew control problem. Unfortunately, unlike the crew scheduling, crew coordinators do not have the benefit and luxury of time, they usually have to solve the problem within minutes—any solution that takes much longer to solve, no matter how good, runs the risk of being useless.

The key in selecting a proper subset of airports in solving the crew control problem is to identify those airports that may contain possible solutions to the problem. The number of airports included should be large enough so that some solutions exist within this region (i.e. the subset of airports selected). On the other hand, the number of airports included should not be so large as to render the problem intractable. Admittedly, it takes experience and intuition to form a proper region. But this is not impossible and there are some rules of thumb that may help the crew coordinator to make wise decision in selecting sub-problems. For example, since the original schedule is built through sub-problem selections, it will be helpful to use the sub-problems formed in solving crew scheduling problem. If we view the crew scheduling problem and the crew control problem from a perspective of "build vs. repair", then it is only natural that we only fix the sub-problem that has trouble.

It should be pointed out that the above approach is most effective for small problems or "minor perturbations", i.e., problems that involve only a few flights or a few airports. For large problems or "major perturbations" such as when a major hub is brought down, it is very difficult if at all possible to identify a reasonably-sized region in order to solve the problem quickly, since the problem generally propagates to the entire system and the recovering process takes much longer than minor perturbation. It is very likely to take a few days to completely recover the system. In this case, the problem becomes much harder and takes longer to solve.

Pairing generation

Like the crew scheduling problem, we need to generate pairings for all the crew members involved in the problem region. These generated pairings should all be legal and can send the crew member in question to his or her return node by the end of the recovering window. Unlike crew scheduling, we can take advantage of the information on original crew pairings. Since we want to minimize the disruption to the original crew schedule, we want to generate pairings that more or less follow the original pairings. This is again a manifestation of the "build vs. repair" strategies.

The properly generated pairings are very critical in solving the problem quickly. The key is to generate enough pairings so that some solutions exist and at the same time avoid generating excessive number of pairings to burden the solving process.

The network in next page is an example that is used to illustrate the OR approach. It is a problem with a region of 4 airports, BOS (Boston), CLE (Cleveland), EWR (Newark) and GSO (Greensboro). There are 18 scheduled flights, 6 regular crews and 1 reserve crew within the problem region. For simplicity, we assume all the flights belong to a single fleet type, DC9, are within the same time zone and also we only consider one crew type (captain, for instance). Table 4.1 lists the flight schedule and Table 4.2 the crew pairings for all the crew members.

To solve any irregular operation crew problem, several parameters are given. The parameter for connection time of swap arcs, i.e. the range of time within which a crew can take a flight from his/her beginning availability time is 8 hours and, the parameter for the range of time before return node that a crew must return is set to be 14 hours. These parameters can be adjusted either beforehand or dynamically in different situations.

In solving any irregular problems from this system, we first generate all possible pairings for each crew using the parameter defined above. In generating pairings, legality checking steps are omitted. Legality checking can be incorporated to eliminate some of the pairings that may be illegal. Since our purpose here is to illustrate the model and the algorithm, it is not considered. A simple cost structure is also adopted. Any crew that takes a flight which does not belong to his/her original pairing incurs a cost of 1.

Table 4.1 Flight schedule of the example problem.

Seq No	Equip	Date	Flt No	Org	Dst	Dep	Arv	Flt Time
1	DC9	940920	1481	BOS	CLE	0730	0930	0158
2	DC9	940920	1519	BOS	GSO	1015	1210	0155
3	DC9	940920	1687	CLE	BOS	0740	0940	0156
4	DC9	940920	786	CLE	EWR	1100	1225	0119
5	DC9	940920	1867	CLE	GSO	1335	1450	0113
6	DC9	940920	1609	CLE	GSO	1650	1805	0112
7	DC9	940920	1568	CLE	GSO	2150	2305	0110
8	DC9	940920	1601	EWR	GSO	0700	0843	0117
9	DC9	940920	1779	EWR	GSO	0830	1015	0121
10	DC9	940920	1690	EWR	CLE	0955	1134	0124
11	DC9	940920	1531	EWR	GSO	1155	1330	0130
12	DC9	940920	1431	EWR	GSO	1300	1440	0136
13	DC9	940920	1626	GSO	EWR	1220	1353	0129
14	DC9	940920	1670	GSO	CLE	1240	1355	0114
15	DC9	940920	1678	GSO	CLE	1545	1700	0108
16	DC9	940920	1591	GSO	CLE	1630	1758	0121
17	DC9	940920	1720	GSO	CLE	1725	1843	0116
18	DC9	940920	1698	GSO	EWR	1825	1957	0130

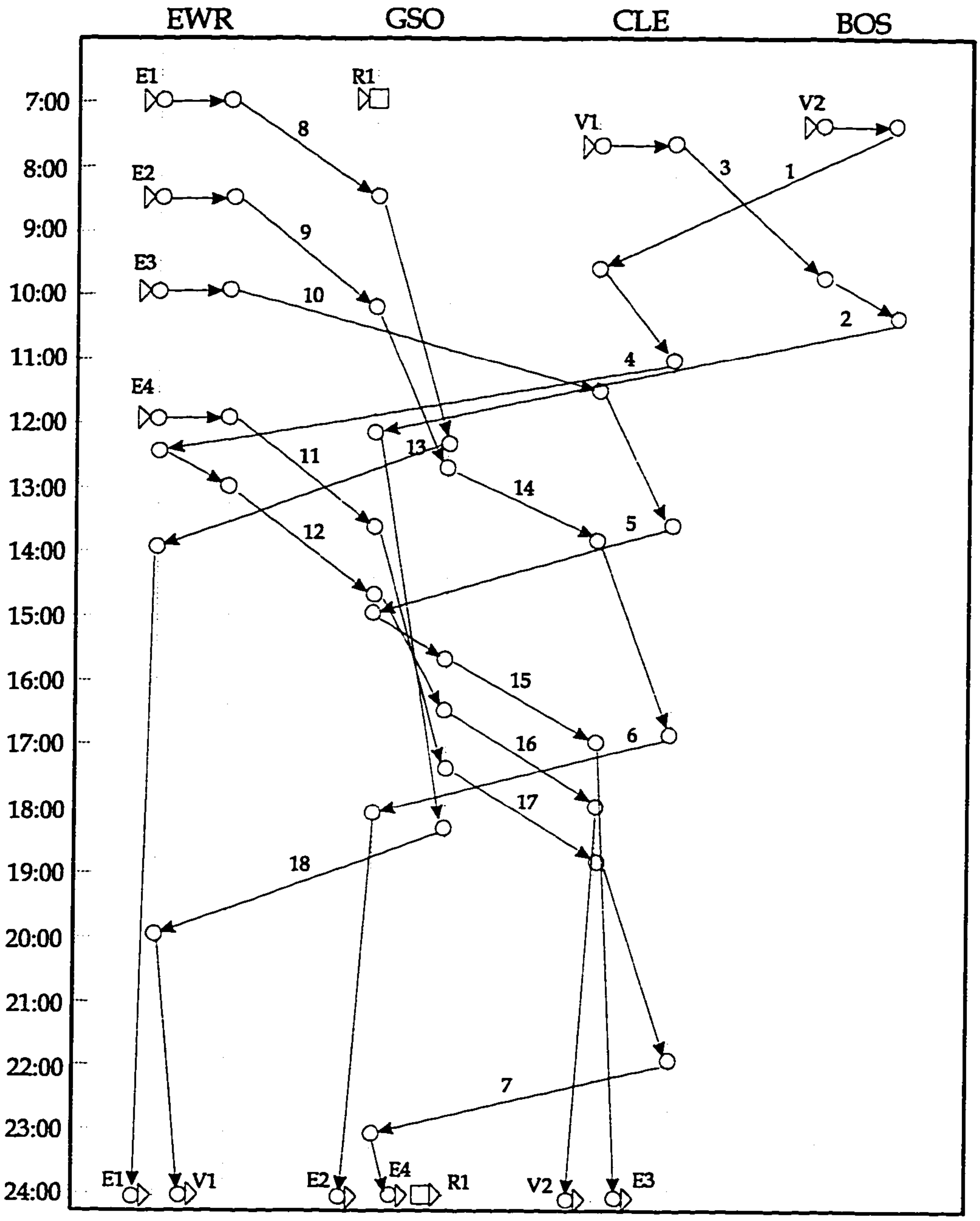

Figure 4.2 The airport-time network for the example problem. The network shows the original flight schedule and crew pairings.

Table 4.2 Crew pairings for the problem in text.

E1								
10	940920	1601	DC9	EWR	GSO	0700	0830	0117
20	940920	1626	DC9	GSO	EWR	1220	1353	0129
E2								
10	940920	1779	DC9	EWR	GSO	0830	1015	0121
20	940920	1670	DC9	GSO	CLE	1240	1355	0114
30	940920	1609	DC9	CLE	GSO	1650	1805	0112
E3								
10	940920	1690	DC9	EWR	CLE	0955	1134	0124
20	940920	1867	DC9	CLE	GSO	1335	1450	0113
30	940920	1678	DC9	GSO	CLE	1545	1700	0108
E4								
10	940920	1531	DC9	EWR	GSO	1155	1330	0130
20	940920	1720	DC9	GSO	CLE	1725	1843	0116
30	940920	1568	DC9	CLE	GSO	2150	2305	0110
V1								
10	940920	1687	DC9	CLE	BOS	0740	0940	0156
20	940920	1519	DC9	BOS	GSO	1015	1210	0155
30	940920	1698	DC9	GSO	EWR	1825	1957	0130
V2								
10	940920	1481	DC9	BOS	CLE	0730	0930	0158
20	940920	786	DC9	CLE	EWR	1100	1225	0119
30	940920	1431	DC9	EWR	GSO	1300	1440	0136
40	940920	1591	DC9	GSO	CLE	1630	1758	0121

Rev Crew	Qualification	Base	Beg Date	Beg Time	End Date	End Time
R1	DC9	GSO	940920	0700	940920	2359

Once the pairings are generated, they are fed into a linear programming solver to solve the mathematical programming problem stated at the beginning of this section. Fortunately, the problem size is not big enough to cause fractional solutions and therefore there is no need to use mixed integer programming solver.

Several cases are created and solved for the problem described above. In the following, flight indices are used to refer to a flight.

Case 1: Cancel flights 4 and 10.

Optimal solution: This is a crew swap solution. Two crew pairings are modified. The other pairings are not affected.

Modified pairings: E3: 12 → 5 → 15

V2: 1 → 5 → 16

Case 2: Cancel flight 5

Optimal solution: Crew E3 will stay at CLE after flight 10, while reserve crew R1 will take flight 15 and deadhead back to GSO on flight 7.

Modified pairings: E3: 10

R1: 15 → 7

Case 3: Delay flight 9 until 13:00, and flight 11 until 17:00. This is a multiple-delay case.

Optimal solution: Crew E2 will take flights 17 and 7, which are missed by E2 now; reserve crew R1 will pick flights 14 and 6 missed by E2; and crew E4 will stay at GSO after flight 11.

Modified pairings: E2: 9 → 17 → 7

E4: 11

R1: 14 → 6

Case 4: Delay flight 5 until 18:00.

Optimal solution: Crew E3 will simply stay at CLE after flight 10, while reserve crew R1 will pick up the two open flights 15 and 5.

Modified pairings: E3: 10
R1: 15 → 5

Case 5: Delay flight 5 until 18:00, without reserve crew.

Optimal solution: The solution involves 3 crew members: E2, E3 and V1. Compared with case 4, E2 will take flights 15 and 5 instead of R1. Flights 14 and 6, which belong to E2 originally, are now served by crew V1. E3 will stay at CLE after flight 10, as in case 4.

Modified pairings: E2: 9 → 15 → 5
E3: 10
V1: 3 → 2 → 14 → 6 → 18

All these problems are solved on a PC486/33. The solution times are around 10 seconds. The linear programming solver used in the program is from "Numerical Recipes in C" [12]. Larger problem is also solved with, for example, 35 flights, 9 pairings, 3 reserve crew. In this case, fractional solutions begin to appear at around 30 percent of the cases. It is expected that the portion of fractional solution will significantly increase with the increase of the problem size. Of course, mixed integer programming solver can be used to resolve fractional solutions. But for problem of realistic size, it is virtually impossible to solve the problem solely based on integer programming. More importantly, because of the business requirements, such as multiple solutions, partial solutions etc., it is almost a hopeless endeavor to use this traditional OR technique to solve crew control problem of meaningful sizes. We thus need to search for alternative algorithms for the problem.

4.3 HEURISTIC ALGORITHM—NON-SPLIT CREW

We have demonstrated the traditional OR algorithm in solving crew control problems and have also discussed some of the major drawbacks of this approach. In this section, we introduce a new heuristic-based algorithm that is more flexible and efficient and can address the issues that were raised for the OR algorithm.

Basic Heuristic-Search Procedure

Before we discuss the new heuristic algorithm for crew control problem, we first introduce some basic concepts and notations about heuristics in general that will be useful in later discussion. We will also look at one particular heuristic-search procedure which are employed by our algorithm. This section follows the discussion in Chapter 2 of Judea Pearl. [13]

Basic Graph-Searching Notation

A heuristic algorithm is often expressed as a graph-searching procedure. A graph consists of a set of **nodes**, which in our context represent the state of the problem. In every graph, there is a special node s called **start node**, representing the initial problem at hand. Certain pairs of nodes are connected by direct **arcs**, which represent operations that can be performed on the node to transform it to a different node (state). If an arc is directed from node n to node n', node n' is said to be a **successor** of node n and node n is said to be a **parent** of n'. The number of successors emanating from a given node is called the **branching degree** (or **degree**) of that node. A pair of nodes may be successors of each other. In this case, the two arcs can be replaced by an undirected **edge**.

A **tree** is a graph in which each node (except one **root** node) has only one parent. A node in a tree that has no successors is called a **leaf**.

A sequence of nodes $n_1, n_2, \ldots, n_k$, where n_i is a successor of n_{i-1}, is called a **path** of length k from node n_1 to n_k. If a path exists from n_1 to n_k, node n_k is said to be a **descendant** of n_1, and node n_1 is called an **ancestor** of n_k.

The most elementary step of the heuristic graph searching algorithm is **node generation**, that is, computing the representation code of a node from that of its parent. The new successor is then said to be **generated** and its parent is said to be **explored**. A different but very important step is **node expansion**, which consists of generating *all* successors of a given parent node. The parent node is then said to be *expanded.*

A **search procedure**, a **policy**, or a **strategy** is a prescription for determining the order in which nodes are to be generated. A distinction can be made about an **uninformed** search or an **informed** search. In the former, the order in which nodes are expanded depends only on information gathered by the search but is unaffected by the character of the unexplored portion of the graph, not even by the goal criterion. The latter uses partial information about the problem domain and about the nature of the goal to help guide the search toward the more promising directions.

The set of nodes in the graph being searched can at any given time be divided into four disjoint subsets:

1. Nodes that have been expanded,
2. Nodes that have been explored but not yet expanded,
3. Nodes that have been generated but not yet explored,
4. Nodes that are still not generated.

Some of the search procedures, such as the depth-first search procedure to be discussed below, require a distinction between nodes of the first and third group (i.e., their successors are available to the search procedure) are called **closed**, whereas nodes that were generated and are awaiting expansion are called **open**. Two separate lists called CLOSED and OPEN are used to keep track of these two sets of nodes.

Depth-First Search Procedure

In **depth-first** search, nodes at deeper levels of the search graph has the priority. Each node chosen for exploration is expanded, i.e. all its successors gets generated, before another node is explored.

After each node expansion, one of the newly generated children is again selected for expansion, the selection of which can be either uninformed or informed. This forward expansion is pursued until a certain criterion is met. This criterion could be a predetermined parameter that decides if the current direction should continue, or it has been decided based on the information accumulated so far that the goal node can not be reached following the current direction, or that the goal node has been reached. If the searching stops, the process can resume from the deepest of all nodes left behind, namely, from the nearest decision point with unexplored nodes. The strategy works well when solutions are plentiful and equally desirable, or when we have reliable early warning signals to indicate if the current candidate direction is incorrect.

In searching trees the concept of depth is well defined. In depth-first algorithm, it is particularly easy to decide which node in OPEN is the deepest. Obviously, the deepest node is the one that is most recently generated. This suggests that in the depth-first search algorithm, OPEN should be structured as a **stack.** During the search process newly generated successors are put on top of OPEN and the next node expansion is the topmost node on OPEN.

For large graphs, it is necessary sometimes to provide a mechanism for recovery from going deeper and deeper along a fruitless path for the depth-first procedure, since the depth of the graph can be almost infinite. To avoid such thing to happen, depth-first algorithm is usually given a stopping rule, **depth-bound**, which, when triggered, gives up the current searching path and starts from the deepest alternatives not exceeding this bound. The procedure thus backtracks under one of two conditions:

1. The depth-bound is exceeded.
2. A node is recognized as a dead end.

The latter occurs when a node fails to pass a test for some property that must hold true for any node on a path to a solution.

We can summarize the depth-first procedure in the following pseudo codes:

Depth-First Algorithm

```
Input: start node;
Initialization: put start node on OPEN;
while (OPEN is not empty) {
    Remove the topmost node from OPEN and put it on CLOSED. Call this node n.
    if ( the depth of n is equal to depth bound)
        clean up CLOSED;
    else {
        Expand n, generating all of its successors;
        Arrange these successors in reverse order in which they are explored. Put them on top of OPEN and provide for each a pointer back to n;
        for (each successor)
            if ( the successor is a goal node)
                get the solution by tracing back through its pointer to parent and exit;
            else if ( the successor is a dead node)
                clean up CLOSED;
    } /* else */
} /* while */
```

The operation "clean up CLOSED" referred about in the above algorithm is performed by purging from CLOSED all those ancestors of the nodes passing the tests in the algorithm that do not have sons in OPEN. This operation is optional and is designed only to save memory space.

Note also that after expanding *n*, we arrange its successors in reverse order in which we want to explore them. This step is optional and depends on whether we have information to give priority to nodes for exploration. The reverse order is due to the fact that once they are put on the stack OPEN, they will be picked in the desired order.

Breadth-First Search Procedure

In the **breadth-first** search, nodes at the same level of the search graph are all expanded before nodes at deeper levels are explored. In some problems, breadth-first search may have advantages over depth-first search.

The above depth-first algorithm can be slightly modified to get a breadth-first algorithm. The most important change is that OPEN is now a queue instead of a stack, which changes the node-searching sequence. CLOSED no longer holds the path from root to current node, so it is not needed. The "clean up CLOSED" operation is thus eliminated. Also, the solution path can be obtained by tracing back through the pointer to parent and the depth of the node can also be calculated similarly.

In the following, we use the depth-first search procedure to illustrate the algorithm. It should be understood that breadth-first search can also be used without any changes to our methodology. Some remarks will be made when there is appreciable difference between the two procedure in terms of performance.

An Example—4-Queens Problem

We use a simpler version of the classical 8-Queens problem, in which one must place eight queens on a chess board such that no queen can attack another, to illustrate the depth-first search algorithm. The 4-Queens problem is a 4×4 board version of the 8-Queens problem. Essentially it requires that no row, column, or diagonal contains more than one queen. Figure 4.3 illustrates the sequence of

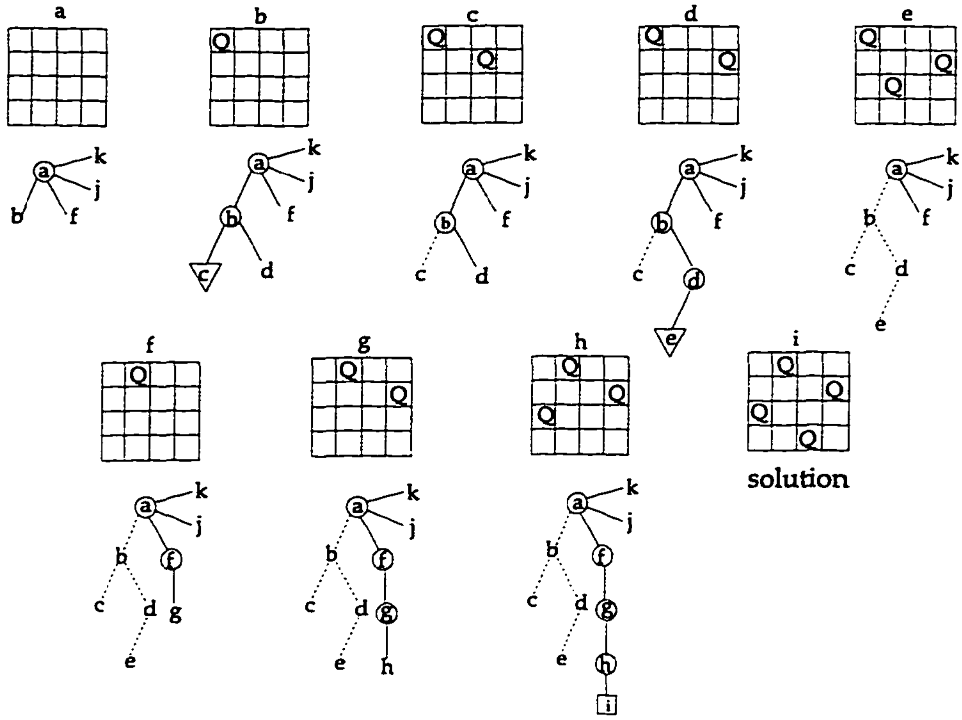

Figure 4.3 Successive steps in a depth-first search of the 4-Queens problem. Circled symbols represent CLOSED nodes, uncircled symbols represent nodes in OPEN, triangles stand for dead ends and boxed nodes are solution nodes.

steps taken by a depth-first search of the 4-Queens problem. Each step is represented by the node being expanded (marked a, b, ..., j, k) and the status of the explicit portion of the search graph after each expansion. The order of nodes on OPEN can be seen by traversing the leaf nodes from left to right, skipping the dotted lines which represent portions of the graph deleted from memory. The order of expansion is further illustrated in Figure 4.4. Note that at any given time the CLOSED list forms a *single* path from the start node to the currently expanded node. This feature reflects the storage economy of depth-first strategies; the maximum storage required cannot exceed the product of the depth-bound and the branching degree. The path of CLOSED nodes maintained by the program appears to traversing the tree, sweeping across it, from left to right (see Figure 4.4) and is called the **traversal path.** Note also that if the "clean up CLOSED" operation is performed in the depth-first search algorithm, then when a goal node (*i*) is found, CLOSED stores the path from start node to the goal node (*a*, *f*, *g*, *h*, *i*).

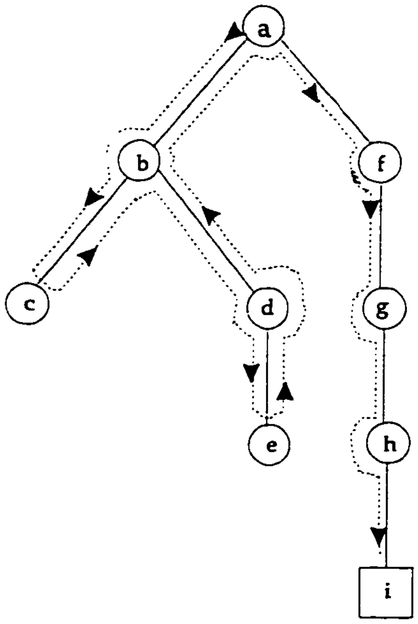

Figure 4.4 Order of node expansion by a depth-first search of the 4-Queens problem.

Heuristic Algorithm for Irregular Operation Crew Management

The biggest problem with traditional OR technique in solving irregular operation crew management problem is that it essentially treats the crew control problem in the same way as it does the crew scheduling problem, which means it still has to generate a huge number of pairings. It in effect ignores the fact that these two types of problems are different in nature. This is what we had called "build vs. repair" strategies. The key in solving crew control problem lies in taking advantage of the current crew schedule. Most of the pairings in current crew schedule are still valid even during severe irregular operations. Therefore most of the pairings generated in solving the problem almost surely will be discarded, thus the waste of time and effort in generating them in the first place. Also, since the pairings are generated in a "batch mode"—they are fed to the integer programming solver in large quantity at a time, it does not provide the information necessary to adjust the pairings generated once it fails to find a solution. In other words, it does not have the capability, if the current effort fails, to backtrack a little bit, modify the existing pairings somewhat and try again in a slightly different path.

Apart from the inherent inefficiency of the above "generate-and-solve" approach, the algorithm developed in last section also fails to address the multiple-solution and partial-solution requirements of the problem. In solving crew problem of irregular operation, often, the most pressing issue is not to find *the* optimal solution, rather it is to find good quality solutions within the time constraint faced by the crew coordinator. In stead of the "all-or-nothing" approach adopted by the OR algorithm, a good partial solution is definitely better than no solution at all. If, during an irregular operation, 90 percent of the hundreds of affected flights are recovered, it is a much better "solution" than a "solution" that can only recover 50 percent of the affected flights, even though all these "solutions" are just partial solutions. The impact to the revenue of the airline is huge.

The above analysis obviously favors a "generate-and-test" approach, in which we generate (modify) one or a few pairings and test the status of the problem, then decide what to do next. The depth-first search procedure fits in here in that it allows us to modify our approaches and backtrack one step to search for a slightly different direction. Even if we fail to find a complete solution within the given time range, we can still end up with a partial solution. Furthermore, depth-first search procedure allows for obtaining multiple solution nodes if we don't stop at the finding of the first solution.

When using depth-first search procedure to solve the crew control problem, a natural question is how efficient the algorithm will be, since the search space is so large that it is impossible to do an exhaustive search. This is where heuristics come in. As can be seen below, we can use the specific business rules and knowledge in developing heuristics to improve the efficiency of the depth-first search procedure.

Experience and Rules of Thumb

In developing heuristics, it is very helpful to familiarize oneself with specific problem domain knowledge in order to gain insight into the business problem. It is also of great help to know and understand the ways people in this particular business solve the problem and the reason behind them. In our case, we have conducted extensive interviews with dozens of operating personnel in different positions as well as their managers in a major North America airline and have also carefully studied all kinds of relevant business rules. Table 4.3 summarizes the main points.

Table 4.3 Summary of the objectives, constraint and the current practices, rules of thumb in solving crew pairing repair problem.

Objectives	• To cover as many flights as possible • To recover the crew pairings as soon as possible
Constraints	All the repaired or reconstructed pairings must be legal
Preferences	Based on the above objectives and subject to the given constraints, preferred solutions are characterized by the following criteria: • The number of modified pairings is minimal • The deviation from original pairing for each modified pairing is kept minimal • Keep crew members originally in the same pairing together as much as possible • Use as few reserve crew as possible • Low cost solution
Current practices and rules of thumb	• Focus on now, solve the more urgent problem first • If necessary, solve the current problem by creating a new problem at a later time ("buy time strategy") • As a result, the priority is on covering the more urgent flights first, with worrying about pairing completion as secondary concern • There is no fundamental difference between solving a small problem and a large problem ("local problem" vs. "global problem") as far as the solution strategies are concerned. A large problem is just made up of many small problems • It is allowable to solve the problem partially, in the sense that a few flights might not be covered and/or a few crew pairing might not be recovered under the current solution • Problem is solved along the line of fleet types

These points are further explained in detail below.

Objectives: In the simplest term, the goal of crew management during irregular operations is to recover the entire system (flights, crew pairings etc.) as soon as possible. If it is not possible to cover all the flights that are disrupted, then it is best to cover as many as possible. The two objectives listed in the table may conflict with each other sometimes. For example, to find crew for a particular flight whose original crew are stuck in an airport that is closed, there may be no other choice than to use other crew with the consequence of disrupting their pairings; or to maintain the integrity of all the crew pairings, the airline may have to leave some flights uncovered. A problem can either be stated as covering flights while maintaining the integrity of crew pairings or repairing damaged pairings while covering flights. Our algorithm uses the former representation of problem.

Constraints: Constraints here refer to the mandatory FARs, company policy and union rules. Constraints must be satisfied by any solutions.

Preferences: Preferences refer to the criteria that can be used to evaluate the quality of the solutions generated. They also provide some guidelines when generating good solutions. The criteria listed in the table are just the broadest and most important measures. There are many other finer criteria that can be used to evaluate solutions. We omit them here. Basically they fall into two broad categories; some of them are management-oriented, e.g. the first three preference criteria, others are cost-oriented. These criteria can be ranked in hierarchical order and assigned different weights if they are on the same hierarchical level. For example, minimizing the number of impacted pairings usually dominates the other considerations and is thus first used to assess the quality of solutions. As will be seen shortly, this criterion is used to prune branches of the depth-first search tree that are deemed to lead to inferior solutions.

Current Practices and Rules of Thumbs: These are some of the heuristic rules crew coordinators use daily to solve crew control problems. These battle-tested rules and practices provide the necessary insight in designing heuristic algorithm. For example, the first rule can be used during the depth-first search procedure to decide which path to follow when there are alternatives. The "buy-time strategy" can propagate the problem to a different airport where there may be more maneuvering space in terms of crew and flights and/or at a later time when more choices are available.

Basic Algorithm

Based on the above analysis as well as general discussions in previous chapters, a basic depth-first search heuristic algorithm can be devised. We first define a generic state representation of the problem which characterizes each node of the search tree. We requires that at each node, the problem be represented by a set of uncovered flights, a list of pairings that are modified so far in the searching process and every pairing is repaired (i.e., no broken pairings), at least temporarily. Obviously, when the set of uncovered flights is empty at a node, that node represents a solution node. At each non-solution node, we pick a flight from the uncovered flight set according to some heuristic rules (e.g., the earliest flight). A candidate crew list is then built to cover this flight from sit crew, layover crew, arriving crew, or reserve crew pool available at the airport. These crew must be available (e.g. they arrive at the airport before the departure of the flight), qualified (e.g., if the flight is a B737, then only crew qualified to operate on B737 can be selected). Different crew from the candidate list will lead to different branches. Thus the operation that corresponds to the arc between a pair of nodes in the search tree is the assignment of a particular crew member or crew members to an uncovered flight. Once the crew is assigned to the flight, either a new pairing must be created for the crew in the case of reserve crew or, the crew's current pairing must be modified due to the rescheduling in the case of regular crew. This is a very important step of the algorithm and is necessary to make the new node a generic state presentation of the problem (no broken pairings). There are several requirements with regard to the newly created or modified pairings: 1) it must be able to send the crew to its designated return node; 2) the pairing must be legal and; 3) the pairing should stick to its original pairing as much as possible. If 1) and 2) are not satisfied, the new node is a dead node. The consequence of this "pairing generation" step is that a few more uncovered flights may be added to the uncovered flight set, since the chosen crew is likely to skip several flight legs in the original pairing. This process can continue from the newly generated node. The stopping criterion can be a predetermined time limit and/or when the number of solutions required has been achieved.

A list of solutions are saved during the search process and are updated whenever a new and better solution is found. The number of solutions required is given before the process starts. A list of partial solutions may also be stored and updated during the process.

There are three important components that deserve separate discussions (see Figure 4.5 for the flow diagram of the solution process). The first one is the preprocessing which converts the initial problem into a generic one. The preprocessing component may involve collecting the uncovered flights and fixing the broken pairings, at least temporarily. The output from preprocessing is the start node for the depth-first search tree.

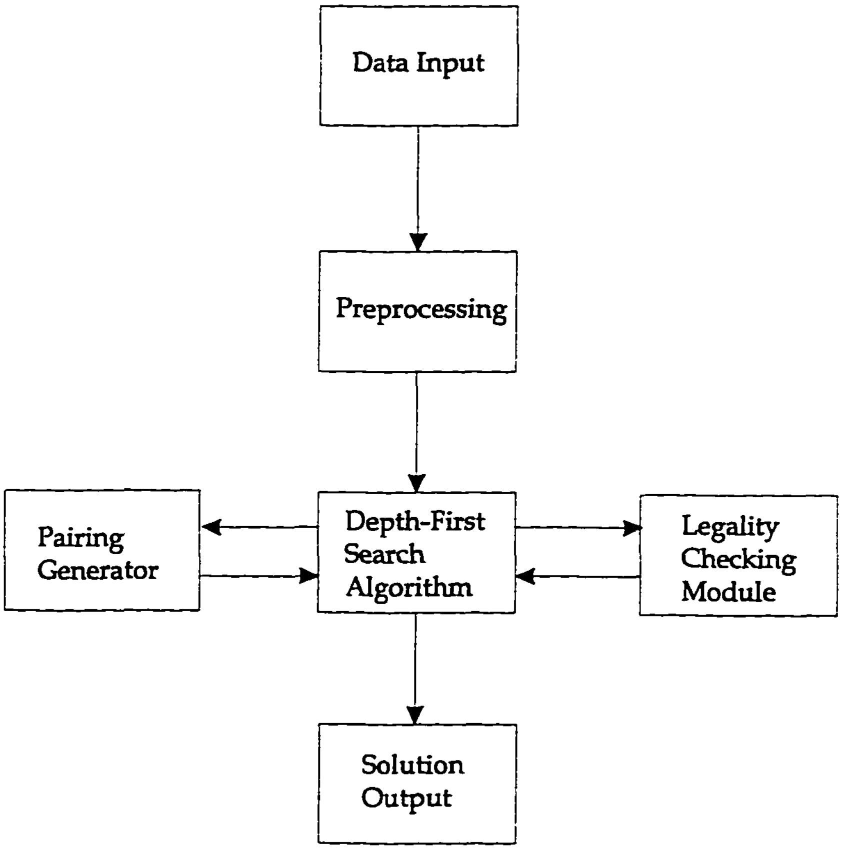

Figure 4.5 Flow diagram of the crew management problem during irregular operation solution process.

The second component is the pairing generation or modification after the assignment of a crew to an uncovered flight. In our algorithm, we use a negative-cost shortest path algorithm to find the trip that leads the crew to its return node. The crew is encouraged to follow the original pairing as much as possible by giving the flight legs in the original pairing negative cost. The negative cost shortest path is possible due to the acyclic nature of the underlying airport-time network. This acyclicity has also enabled us to use more efficient version of shortest path algorithm. [14]

The third component is the legality checking module that is invoked after the pairing generation or modification. The legality checking module is used to check if the pairing is legal. It is an independent module and is intentionally designed to be so. Since the FARs, company policy and union contract may change from time to time, it is critical that these changes do not affect the rest of the program.

To improve the efficiency of the algorithm, a pruning scheme is necessary to avoid searching for branches that will not lead to better solutions than are already found. We take advantage of the fact that the number of modified pairings are the foremost dominating factor in deciding the quality of a solution as well as the fact that the number of modified pairings is non-decreasing along any search path, or the path from root to current node. The number of modified pairings is the same at the current node as the parent node when the crew assignment/reassignment leading to this node is for one of the already-modified pairings; it is incremented when the assignment/reassignment is for an original pairing. Therefore, if at any node, the number of modified pairings is larger than that of the solutions found so far, then there is no need to explore the node. The note is designated as infeasible. Figure 4.6 presents the algorithm.

```
preprocessing and get the start node;
put start node on OPEN;
max_pair = ∞;   // largest number of pairing modified so far, used as bound
while (OPEN is not empty) {
   remove the topmost node from OPEN and put it on CLOSED, call it n;
   if (depth of n is equal to depth bound)
      clean up CLOSED;
   else {
     select a flight f  with earliest departure time in the uncovered flight set of n;
     build a candidate list L of crew who can serve f;
     expand n, generating all its successors; each successor represents a crew on L serving f;
     put these successors on top of OPEN, in reverse order in which they are to be explored;
     for (each successor s) {
       if (s is infeasible)
         clean up CLOSED;
       else if (s is a solution node) {
          remove s from OPEN;
          if (number of solution found so far < required number of solution NS)
             save solution s and clean up CLOSED;
          else
             update solution list S with s and max_pair, clean up CLOSED;
       }
       else      // s is not a solution node
         if (not enough solution found)
            save s in or update with s the partial solution list P;
     }/* for */
   }/* else */
}/* while */
```

Figure 4.6 Algorithm for non-split crew.

It should be pointed out that so far the algorithm only deals with crew on a flight as a single unit. While this is the way the crew pairings are built during crew scheduling stage and up to 80 percent of the pairings fall into this type, there are still pairings which may contain split crew. Thus the pilots and flight attendants may belong to different pairings even though they serve the same flight. Even worse, although it is rare, there may be cases where every individual crew member on a flight assumes different pairings. This situation complicates the solving process. We will discuss this and other issues later on. At this point, we just assume that all crew are non-split.

Computational Results

The heuristic algorithm is applied to the same example problem in Figure 4.2. The program is written in C++ and is developed on HP 715/100. The results are summarized in Table 4.4. The numbers in the brackets are the actual number of solutions generated.

From the computational results in Table 4.4, several observations can be made:

It takes the program only a few nodes and a very short time to obtain the first solution. This is a very important result.

The pruning scheme is very effective when there are large number of solutions. It can reduce around 50 percent of nodes that need to be generated. For example, in case 2, if we need only one solution, the number of nodes generated is only about 30 percent of all the nodes that can be generated if we do not limit the number of solutions to be generated (96 out of 330).

It does not take much additional time to obtain more solutions. This can be attributed to two reasons. First, it takes considerable searching to get to the optimal solution, by which time there are many solutions that have already been generated; Second, in the problem we studied, the difference between solutions in terms of number of pairings affected is not very large, resulting in many solutions very close in quality. This tend to reduce the effectiveness of pruning. In fact, many solutions obtained are almost the same, only differing by one or two flight legs.

Table 4.4 Summary of computational results from heuristic algorithm

Case No.	Time taken to obtain first solution (*sec.*)	Number of nodes generated to obtain first solution	Number of Solutions Requested	Number of Nodes Generated	Solution Time (*sec*)
1	1.88	23	1	78	5.39
			3	111	7.68
			1000 (34)	131	8.88
2	0.93	8	1	96	7.08
			3	160	11.81
			1000 (122)	330	24.22
3	0.74	7	1	8	0.79
			3	8	0.79
4	0.72	7	1	54	3.72
			3	66	4.49
			1000 (36)	96	6.50
5	0.6	4	1	6	0.72
			3	6	0.72
			1000 (3)	6	0.72
6	0.79	7	1	54	4.01
			3	54	4.17
			1000 (3)	96	7.17

Preprocessing is very critical in determining the efficiency of the program, since preprocessing generates the start node. A good start node can quickly lead to a solution or optimal solution, thus reducing the number of nodes that need to be generated.

4.4 HEURISTIC ALGORITHM—SPLIT CREW

The algorithm developed so far only applies to non-split crew. But it can be extended to split-crew with some modifications. In this section, we discuss the algorithm for split crew and the computational results based on the algorithm.

The Algorithm

In Chapter two, we have discussed that pilots and flight attendants have different legalities and work rules. While pilots are generally only qualified to operate on one fleet type, flight attendants usually do not have such restrictions and can operate across equipment types. We have also discussed that although airlines' make great effort to build pairings that are non-split, because of the complicated network and crew legalities, it is almost unavoidable to have split-crew pairings. Indeed, in some cases, every individual crew member serving a particular flight may belong to a different pairing. Furthermore, during the execution of the flight and crew schedules, irregular operations can also disrupt an otherwise non-split crew pairing. Although the majority of the pairings are non-split, it is fairly common for airlines' to split crew pairings to solve an irregular operation problem on hand, which may not be easily solved without splitting crew pairings.

Based on the above arguments, we can naturally decompose a problem according to different crew positions, i.e., captain, first officer, second officer and flight attendants, depending on the required positions on a given fleet type. Table

4.5 lists the crew complements for some fleet types based on the interview with the airline.

Table 4.5 Crew complements for various fleet types

Fleet Type	**B727-all**	**B737-100/200**	**B737-300**	**B737-500**	**DC9**	**MD80**	**B757**	**B747**	**DC10**
CA	1	1	1	1	1	1	1	1	1
FO	1	1	1	1	1	1	1	1	1
SO	1							1	1
IRA						*	*		
FA	3	3	3	3	3	3	5 - 6	14 - 17	10

*May have an IRO, or International Relief Officer.

From Table 4.5, it is clear that no matter how many positions a particular fleet type requires, each pilot position needs only one crew member of the type, whereas the number of crew members required for flight attendant position varies from fleet type to fleet type. The latter point raises the question of whether flight attendant position should be further decomposed into individual crew members. For instance, if we have a 3-crew flight attendant pairing, should we decompose it into different combinations of subgroups? In other words, do we want to decompose the group into 1+2, 1+1+1 subgroups. Although it can be argued that since each individual crew member may and indeed have different accumulated flight/duty times or other legality status and it can provide more options if split flight attendants are considered and thus enlarge our solution space, from a practical standpoint, we decide it is generally not a good idea to intentionally split the flight attendants that belong to the same pairing originally. The pairing may be split as the result of irregular operations (one of the flight attendants suddenly becomes sick, for instance).

The arguments against flight attendant splitting are as follows: First, it is preferable that crew in the same pairing stay together as much as possible both because the crew themselves do not want their schedules to be disrupted and the administrative reasons, as discussed previously; Second, and this is the primary reason, once we begin to allow each individual flight attendant to be reassigned and regrouped, the number of different combinations would be enormous, considering the number of flight attendants involved. This would severely affect the efficiency of the program and the performance of the system overall; Finally, as has been discussed previously, flight attendants are less expensive (thus it is more affordable to have reserve flight attendants), their work rules and legalities are less restrictive, and they can operate on different fleet types. All these mean that even if we do not allow further splitting of flight attendant pairings, there are still many options when it comes to reassign flight attendants. What we lose in the solution space and flexibility, we gain at the efficiency and performance of the system. This argument is supported by our computational results given later.

Once we decompose a problem into its different crew positions, we can apply the basic algorithm for non-split crew to each position and thus solve them almost as independent non-split sub-problems and then combine the solutions for individual position to form a complete solution. The complication here is that we need to consider the connection between crew members on different positions when they belong to the same pairing to begin with. Indeed, the vast majority of pairings falls into this category. As is mentioned before, it is preferable to have crew in the same pairing stay together as much as possible, and if they are reassigned, they had better be reassigned together. Our algorithm provide mechanism to take into consideration this preference. The algorithm solves each sub-problem position by position: captain, first officer, second officer (if any), and flight attendant, in that order. In solving first officer sub-problem, we first check if the first officer belongs to the same original pairing with some captain, if so, his preferred pairing will be the pairing of that captain even if the captain's pairing may have been modified when solving captain position. For second officer, if he starts in a pairing which includes a captain, his preferred pairing is always the captain's whether the pairing also includes a first officer or not. Only in the case where the original pairing includes a first officer but not a captain, will his preferred pairing be the one of the first officer. In other words, the lower rank crew will try to follow the highest rank crew in his original pairing once they split due to irregular operations. The complete algorithm is presented in Figure 4.7.

```
solve captain position using the basic heuristic algorithm;
for (each captain solution) {
  preprocess for first officer position, copy solution from captain if possible;
  solve first officer solution using basic heuristic algorithm;
  for (each first officer solution) {
     preprocess for flight attendants, copy solution from captain/first officer if  possible;
     solve flight attendant position using basic heuristic algorithm;
     save/update the final solution list;
  }
}
```

Figure 4.7 Algorithm for split-crew.

The algorithm in Figure 4.7 assumes there is no second officer position. The second officer position can be easily added after solving first officer. The basic algorithm for solving flight attendant position is somewhat different from that of solving other positions. The difference lies in the fact that a single flight attendant pairing may have different number of flight attendants. Remember in the basic heuristic algorithm, a candidate list is formed from the qualified crew pool to serve a flight whose position under consideration is uncovered. In this case, every candidate in the list is an individual. But in the case of flight attendant, the flight which needs flight attendants may have different number of flight attendant positions uncovered. Therefore we may not form a candidate list of flight attendants for a particular flight directly by putting qualified flight attendant pairings on the list, since the number of flight attendants in a pairing may not match the number of flight attendants needed by the flight to complement the flight attendant position. To solve this problem, we can modify the basic algorithm's candidate selection procedure: Each candidate in the candidate list may be a combination of a number pairings such that the sum of the number of flight attendants in each pairing of the candidate is at least equal to the number of flight attendants needed by the flight under consideration. For example, Flight 187 needs 3 flight attendants to complement its crew. We can form a candidate for the flight by using a 3-member flight attendant pairing, a two-member pairing plus a single-member pairing, or, in the extreme case, 3 single-member pairings.

To summarize, the moral of the split crew algorithm is that we allow crew pairings to split along the position (captain, first officer, ...) with a string attached—we preferred the lower-ranking position to follow their higher-ranking crew, but we do not allow split of different flight attendants within the same pairing.

Computational Results

The above algorithm is applied to a number of problems of different sizes. The first problem we tested has 6 airports, 51 flights in a two-day period, 18 pairings including 4 non-split pairings, and 30 individual crew members. The crew complement for this fleet type is one captain, one first officer and three flight attendants. 14 of the 18 pairings are split crew in various ways. The proportion of split crew pairings is much higher than it normally is. It is purposely designed to be so to test the algorithm. The number of airports and flights involved are also typical of most irregular operation problems, particularly minor perturbations. Table 4.6 gives the computational results for different cases created from this problem.

Table 4.6 Computational results for a 6-airport, 51-flight problem.

Case ID	Captain (sec.)	First Officer & Flight Attendants 1	2	3	Number of Solutions
C202	0.72	0.01	52.65	11.77	1
C203	0.63	0.01	0.36	0.37	3
C204	0.42	0.01	0.36	1.32	4
C205	0.33	0.01	25.6	3.29	9
C206	0.29	0.01	0.33		2
C207	5.76	0.01	76.07	6.02	10
C208	2.97	11.14	9.86	27.49	16
C209	1.37	14.79	0.00	27.49	3

The problem was run on a HP9000/K420 parallel system with 2 processors. In this case, only one processor is used. The table lists the run-time's it takes to solve the captain position and the run-time's it takes to solve first officer and flight attendant positions for each of the three best captain solutions obtained during captain-solving stage. In the cases where the run-time's are close to zero (e.g., 0.01), it usually means the solution is obtained by directly copying the captain solutions. As pointed out earlier, the problem we tested are designed to contain higher-than-usual percentage of split crew. In practice, we would expect much more direct copying cases, further enhancing the performance of the algorithm. In some cases, it takes much longer than average to solve certain positions, as in solving the first officer and flight attendant positions in the second captain solutions of case C202 and C207. Still these run-time's are acceptable. The last column in the table records the number of full solutions for all positions obtained for each case. In all the cases tested, we limit the solver to get at most the best three solutions for each position. In some cases where we can easily solve the captain position during preprocessing stage, thus can only get one captain solution, we perturbed the solution using some heuristics to get multiple solutions (up to three). In case C206, we only get two captain solutions, so only two run-time's are listed under the columns of first officer and flight attendant.

4.5 HEURISTIC ALGORITHM—SWAPS

In the last two sections, we have developed a search-based heuristic algorithm that can solve reasonably large-size problems that airlines encounter most of the time. Still, in solving very large-scale irregular operations problems, such as when one or several hubs are brought down, it may take the algorithm an exceedingly long time to find a solution. In this section, we introduce yet another heuristic algorithm that can solve major perturbation problem very fast.

It should be noted that the previous heuristic search algorithm is very generic, in the sense that it does not take advantage of many properties of the underlying problem. All it requires is that the network be acylic. In addition, several characteristics of the algorithm should be carefully examined:

1. The algorithm covers uncovered flight legs one by one. It is thus inherently near-sighted.
2. The algorithm considers all the crew in the system as potential candidates to cover any uncovered flights. As a result, many crew whose pairings are not broken are impacted undesirably.
3. The shortest path algorithm used to send a crew from one point to another is called so frequently that it becomes the performance bottleneck.

There are drawbacks associated with each of the above characteristics. Let us analyze one by one. During an irregular operation, a broken crew often has to skip *a sequence* of flight legs. For instance, a pairing starts from Houston may have the segments: (IAH, AUS), (AUS, DFW), (DFW, LAX), (LAX, SFO), (SFO, LAX), (LAX, IAH). If, due to weather problem, the leg from IAH to AUS is canceled, the crew is rescheduled to either deadhead or fly another flight from IAH to LAX to catch up with its pairing. In this case, a sequence of two legs: (AUS, DFW), (DFW, LAX) is skipped. In the heuristic search algorithm, we would cover the leg from AUS to DFW and the leg from DFW to LAX separately, although the more desirable and also more efficient way to cover them is to use a single crew to cover them as a whole. This suggests that the better way to cover flight legs is to skip and cover flight segments as a sequence instead of treating them individually.

The second characteristic of the search algorithm means that the search space includes all the crew currently in system, including both broken crew and unbroken crew. Although this implies possibly more and better solutions, it also tends to increase searching time, even if a bounding scheme is provided to cut down the number of impacted pairings. For very large-scale problem, such as when one or several hubs are closed, the response time of the algorithm may be too long. This suggests that if we limit the use of unbroken crew as much as possible right from the beginning, we can significantly cut down the search space and thus improve response time. Of course, this calls for judicious rescheduling of broken crew.

Finally, with regard to the third characteristic of the search algorithm, a more efficient and flexible problem-specific heuristics can be adopted to replace the shortest path algorithm. Obviously, in the shortest path algorithm, one has to find the shortest paths from one particular point in the network to all the other points in order to find the shortest path between that point to another particular point. This will become quite inefficient when a large network is searched and the shortest path algorithm is called frequently. In addition, considering the legality constraints, the shortest path algorithm is not very flexible. Once the path found by the algorithm violates the legality requirements, the particular search step will be considered infeasible whereas in practice, an alternative path will do.

The heuristic algorithm that will be developed in this section is a swap-based heuristic. It will improve upon the previous search algorithm, especially the drawbacks we have just listed. Swaps are powerful ideas in solving many combinatorial problems, as was demonstrated by the classic work of Kernighan and Lin [16] in solving graph partitioning problem and Lin [17] in the Traveling Salesman Problem (TSP).

Classification of Solution Patterns

We begin by identifying some conceptually simple yet very powerful solution patterns. We start by looking at fixing one or a few broken pairings.

Balanced Cancellation

Figure 4.8 represents a situation in which two consecutive flight legs in a pairing are canceled: Flights from EWR to TPA and from TPA to EWR. This happens probably because of the decision made by operations manager to balance the aircraft flow.

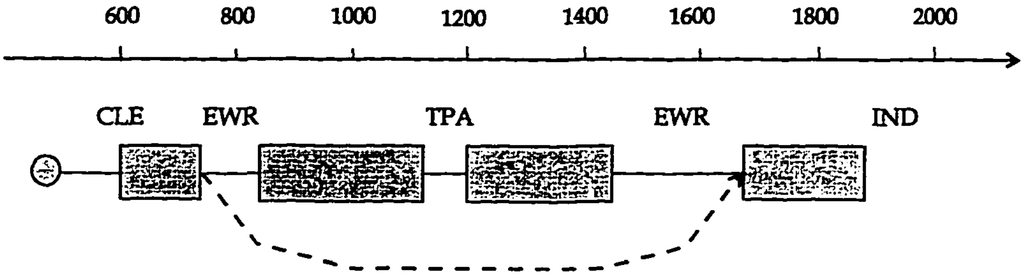

Figure 4.8 Balanced cancellation.

In this case, the crew is automatically balanced because it remains at the same airport where next unperturbed flight in its pairing departs. This is the simplest solution pattern for a particular broken pairing. We define it as the **order 0** solution.

Deadheading

Figure 4.8 shows the situation in which one or more flight legs in the pairing are either cancelled or skipped, but the crew can deadhead on other flight(s) to get the his/her destination either returning his/she base or catching the next unperturbed flight in the pairing. The difference with balanced cancellation situation is that the broken pairing is not automatically balanced, but that the affected crew needs to deadhead on other flights to fix the pairing.

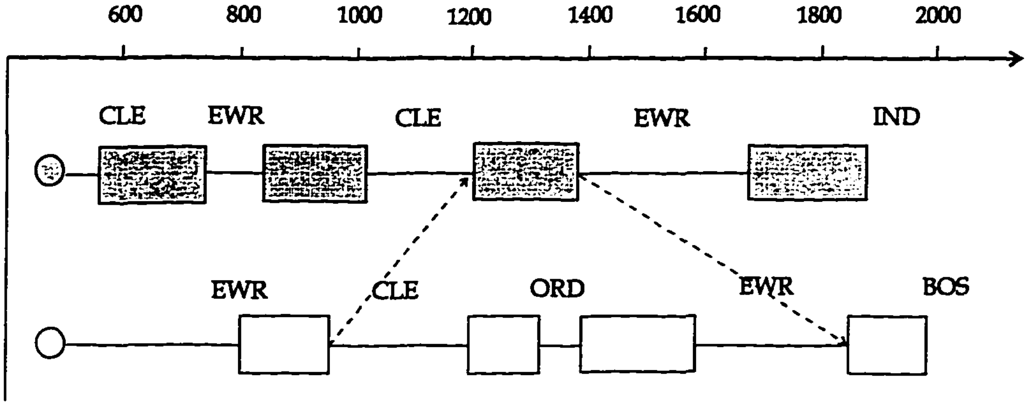

Figure 4.9 Deadheading.

In this case flights from CLE to ORD and from ORD to EWR are cancelled. The crew is stuck at CLE but needs to get to EWR. He/she deadheads on a flight from CLE to EWR. We define the solution pattern for using deadheading in such way the **order 1** solution.

Two-Way Swap (Spatial)

Figure 4.10 illustrates a more complicated situation in which two pairings are broken due to cancellations. In the first pairing, flight leg (CLE, EWR) is canceled; In the second pairing, flight leg (EWR, CLE) is canceled. By swapping sequence of flight legs (EWR, TPA), (TPA, EWR) in the first pairing with the sequence of flight legs (CLE, ORD), (ORD, EWR) in the second pairing, each pairing is now repaired. This example shows a spatial two-way swap because the two pairings both have discontinuity in space, i.e., they can not get to their respective destination airports before swap.

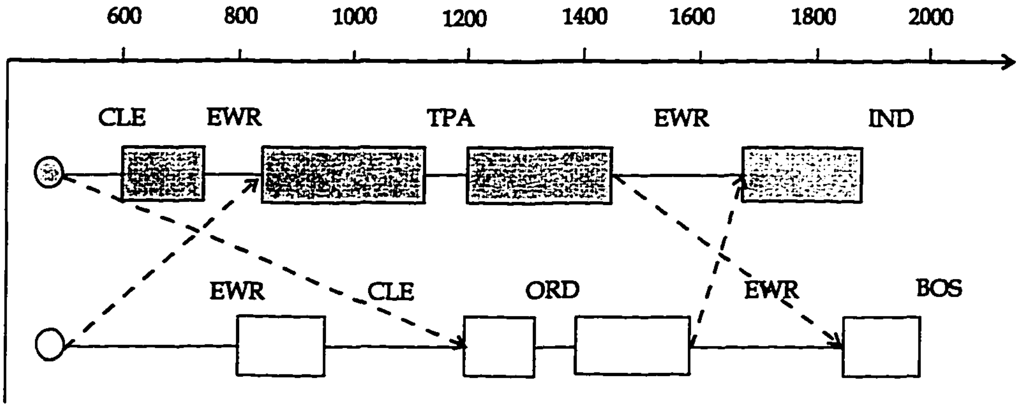

Figure 4.10 Spatial two-way swap.

Figure 4.10 demonstrates a very "clean" two-way swap, in the sense that both crew assigned to the pairings are at the "right place" and "right time" so that the swap is feasible. In practice, there are various variations of this "clean" version of two-way swap. For instance, it may take a few deadheading legs for either or both of the crew to get to the right airport to start the swap, and it may also take deadheading legs for either or both of the crew to get to the right airport to complete the swap. These less "clean" swaps are also less desirable due to the non-productive deadheading legs, so they may be assigned higher costs.

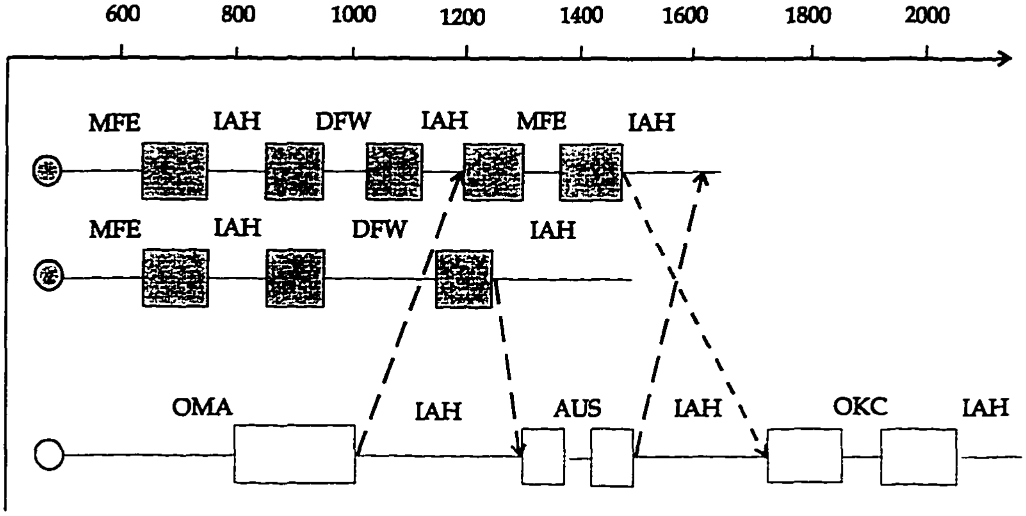

Figure 4.11 Temporal two-way swap.

Two-Way Swap (Temporal)

So far, only broken pairings due to flight cancellations are considered. But we will show that broken pairings due to flight delays can also be handled in a similar fashion using swaps. Figure 4.11 illustrates a situation in which the flight leg (DFW, IAH) in the first pairing is delayed by about one and half hours such that the crew will not be able to catch their next flight leg (IAH, MFE). The crew is at the "right place" but not at the "right time". This broken pairing can be repaired by introducing an appropriate unbroken pairing and swapping sequences of flight legs between them. In this example, an incoming crew to IAH with right arrival time can take the sequence of flight legs (IAH, MFE), (MFE, IAH) from the broken pairing and skip its sequence of flight legs (IAH, AUS), (AUS, IAH), which will be in turn taken by the crew on the broken pairing. They both return to their respective original schedule after the swap.

Like spatial two-way swap, various variations of this relatively "clean" version of temporal two-way swap can be introduced with the help of deadheading flight legs. We also define both types of two-way swap as **order 2** solution.

Higher-Order Swaps

Similarly, the above two-way swaps can be extended to even more complicated solution patterns, in which three or more pairings are involved. In a three-way swap among pairings A, B, C, for instance. A may take a sequence of flight legs from B; B a sequence of flight legs from C; C, in turn, takes a sequence of flight legs from A. Four-way swap and even higher-order swaps can be similarly defined. We also define these solution patterns as order 3 solutions, order 4 solutions, etc.

Path Finding in a Hub-and-Spoke Network

We have seen so far that, in order to search for various possible order 1 and higher solutions, we need to find a sequence of flight legs from one airport to another frequently. In the heuristic search algorithm we developed in section 4.3 and 4.4, we use a generic shortest path algorithm for an acyclic network. As is discussed at the beginning of this section, this algorithm, although very efficient used separately, will become very inefficient and inflexible in our program since it is called so frequently.

We can design a much more efficient and flexible heuristic algorithm to find a relatively short path between two airports by taking advantage of the characteristics of hub-and-spoke network adopted by almost all the major carriers in the U.S. The idea is that since each airport must be connected to at least one hub and there are connections between every two hubs, then there must exist a path between any two airports within a hub-and-spoken network such that the path is composed of no more than three legs. Since any major carrier has at most a few hubs, it takes significantly less search time when finding the path than with the generic shortest path algorithm on the same network. Furthermore, since we do not necessarily need to find the shortest path and often the shortest path may not be a legal one anyway, the heuristic algorithm can easily find alternative paths between two airports.

Decomposition of Solutions

In this subsection, we will discuss the heuristic algorithm that is solely based on the above solution patterns. In other words, we would like to decompose a complete solution to an instance of the crew pairing repair problem into a series of order 0, order 1, and order 3, ... solutions (see Figure 4.12). Our empirical experience shows that, even if it is not guaranteed theoretically that a full solution will be found this way for every instance, we can find full solutions or very good partial solutions for all of the very large problems. In fact, it is our belief that such an algorithm is the best way to solve irregular operations crew pairing repair problems since it provides good quality solutions with quick response time for even very large-scale problems. The idea of swapping sequences of flight legs among pairings also has the appeal of, to a very large part, preserving the efficiency and legality built into the original schedule.

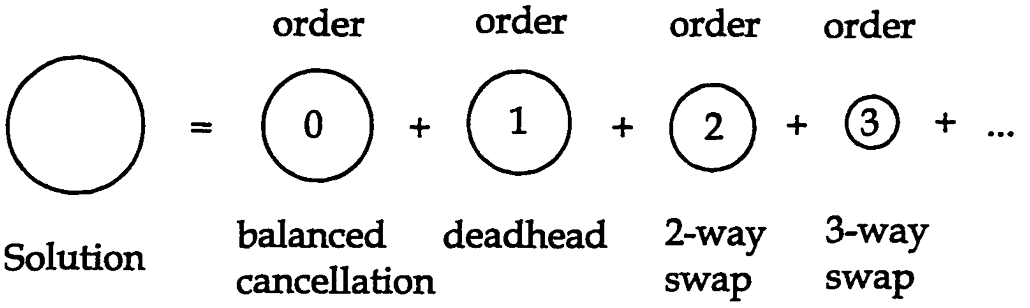

Figure 4.12 A full solution can be decomposed into solutions of simpler patterns.

The main idea of the algorithm is that, for each broken pairing, we try to search for solution patterns of different orders for this particular pairing and save them. We then match the broken pairings with their possible solutions so that overall, maximum number of broken pairings will be fixed. In practice, it may be unrealistic and/or not worthwhile to search for solution patterns higher than order 3. In fact, as will be shown in the following, solution patterns up to two-way swaps are sufficient to solve even very large problems. This surprising result can be compared rather interestingly with the heuristic algorithm for TSP using 2-opt, 3-opt, 4-opt [17]. For the TSP, 3-opt solutions are much better than 2-opt solutions, but 4-opt solutions are not sufficiently better than 3-opt to justify the additional running time. Of course, our problem is quite different from TSP and the very definition of solutions is also different. But this association is certainly enlightening.

Computational Results

The following two tables (Table 4.8 and 4.9) show the computational results for 4 testing cases. The data are provided by Continental Airlines and are the real flight schedule, pairing and crew data. All four cases are for major perturbations. At the time of this writing, Continental has three domestic hubs located at Houston (IAH), Newark (EWR) and Cleveland (CLE), with daily departure flights around 300, 200, 100, respectively. B737-300 is Continental's largest fleet type in terms of daily departures.

Table 4.7 Case descriptions

Case No.	Fleet Type	Daily Departures	Canceled Flights	Notes
C1010	B737-300	297	28	CLE closed for 4 hours
C1014	B737-300	297	49	EWR closed for 4 hours
C1114	B737-300	297	56	EWR and CLE closed for 3 hours
C1216	B737-300	297	57	CLE closed for 11 hours

Table 4.8 gives the computational results for the four cases listed in Table 4.7 using only solution patterns up to two-way swaps. In searching for complete solutions for the problems, we start with a set of broken pairing and do an exhaustive search for solution patterns of up to order 2 for each broken pairings and record the results. A branch-and-bound search procedure similar to the one we discussed in previous section is adopted to search for complete or good partial solutions. At each search node, a broken pairing is selected for repair. A list of successor nodes are generated by committing this pairing to its different solutions

in its solution candidate list. If the broken pairing selected has empty solution candidate list, put it in the uncovered sequence list. In either case, the broken pairing is deleted from the broken pairing set. If this particular solution is a two-way swap, then the other pairing is also deleted from the set if it is a broken pairing. For branching, the broken pairing with least number of alternatives is selected. The initial bound is set to the total number of broken pairing. When the initial broken pairing set is empty, it is updated to the number of uncovered sequences if that number is smaller. The search stops when a given number of solutions are found or a predetermined time limit is reached.

Table 4.8 Computational results using solution patterns with orders less than 3.

Case No.	Broken Pairings	Order 0	Order 1	Order 2	Number of Solutions	Solution Time (sec)
C1010	20	4	9	7	>40	242
C1014	32	7	12	13	>40	149
C1114	33	12	8	13	>40	584
C1216	34	6	17	11	>40	106

Table 4.9 Computational results using solution patterns with any orders.

Case No.	Broken Pairings	Order 2	Higher Orders	Uncvd. Seq.	Number of Solutions	Solution Time (sec.)
C1010	20	10	7	1	3	2
C1014	32	8	13	3	3	15
C1114	33	10	13	3	3	5
C1216	34	10	11	2	3	5

Instead of using only solution patterns up to two-way swaps, we can also use higher order solutions to solve instances of problems. But instead of searching specifically for possible 3-way or higher order solutions, we let the instance of the problem to decide what solution patterns will eventually be used in the final solution. For each broken pairing, we still find out if there exist order 0, order 1 and "clean" order 2 solutions and record the solutions. By "clean", we mean the two-way swaps can be started without using deadheading legs, although we allow them to finish the swapping by using deadheading legs. The rationale behind this is that "clean" two-way swaps are high quality solutions and it is desirable to implement them.

For those broken pairings without solutions of orders lower than 3, a similar strategy to the one used in the problem-general search heuristic algorithm in section 4.3 can be adopted to search for potential higher order solutions. The difference is that now we take or skip sequence of flight legs instead of a single flight leg at a time. The advantages are that we now have a longer view compared with the more local and shorter-sighted view in previous search algorithm and that we cover uncovered flight legs by sequences instead of by individual legs. This will be very useful when reserves are needed to cover flights. For instance, if there are three uncovered flight legs and they do not form a sequence, we will need three reserve crews to cover them whereas if they are a sequence, we only need one reserve crew. The result of this algorithm is that different pairings may end up swapping sequences of flight legs among themselves and the order of the swaps we can not decide beforehand. The disadvantage of the algorithm is that we may end up with sequences of flight legs no crew can cover. Of course, we can use reserves to cover them. But there are two restrictions associated with using reserves which must be carefully taken into account. First, reserves are not available at every airport, but are concentrated at a few crew bases. We can, of course, deadhead reserves to airports where they are needed; Secondly, in most cases, reserves are not available immediately. They are only available during certain time intervals and it takes a few hours of "prenotification" time before they can start to serve any flights. These two restrictions require that there be sufficient time before now and the time reserves will be available. In other words, the uncovered sequences of flights should be preferably at later times. In practice, crew coordinators try to use reserves only from the second day of irregular operations. Obviously, there are exceptions to these rules. In our algorithm, we try to have the sequences of flight legs end at one of the hubs, since at hubs there are more flights and it is more likely for the crew who are taking this sequence to get back to their original pairing or for the reserves who are serving this sequence to get back to their base.

Table 4.9 gives the computational results for the same cases in Table 4.7 based on the above algorithm. It is clear from the table that the run time performance of this algorithm is much better than the algorithm using only solution patterns lower than order 3. But it is also possible to have a few uncovered sequences in the final solutions. One interesting comparison is that the number of order 2 solutions in Table 4.9 is actually very close to that in Table 4.8. This suggests that most of the order 2 solutions are "clean" type and that most of the time in the previous algorithm is spent on exhaustive search of all possible order 2 solutions.

It is possible, however, that we may end up with sequences of flight legs that can not be covered by either duty crew or reserves crew. The remedy for this scenario is doing some postprocessing so that the sequence can be postponed further until there is sufficient time to call up reserves.

In conclusion, we presented several algorithms for solving crew pairing repair problem in this chapter. We pointed out that heuristic algorithms are the only feasible approach to solve large-scale instances of crew pairing repair problem. We introduced two heuristic algorithms: a problem-general search algorithm and a problem-specific swapping algorithm. The former has the advantage of much larger search space with the drawback of possible degrading performance as the size of the problem grows. The latter has much better performance and can be implemented for real-time decision support for problems of major perturbations. The link between the two is that the latter adopts a similar search structure and algorithm to the former. Our problem-specific heuristic algorithm bears some similarities to the famous Feynman's diagrams in quantum field theory. The association comes from the fact that in both cases, a "perturbation" to the "system" has occurred and a complete solution can be decomposed into solution patterns of increasing complexity. Additionally, both Feynman's diagram and our system are based on space-time coordinate.

Chapter 5 A Decision Support System for Crew Management

In this chapter, we propose a decision support system environment for the crew management in irregular operations. This environment takes advantage of the state-of-the-art information and computing technology as well as integrates seamlessly with the legacy system that exist in major airlines.

The following factors should be taken into account when designing the decision support system:

The decision support system will be operating in a heterogeneous environment. All the major airlines have invested heavily in many mainframes and their applications. In the foreseeable future, these legacy systems will continue to exist. Meanwhile, new applications will be developed on powerful and more cost-effective workstations and personal computers. The decision support system must be integrated seamlessly into this environment

The decision support system should be able to handle transaction intensive processes. The operating environment is very dynamic and likely deals with hundreds of thousands of messages per day.

The decision support system should be able to handle computational intensive process. As have been discussed in last chapter, the algorithm will be used to solve large-scale problems that are CPU intensive.

The decision support system should be user-friendly and provides various functionality to aid crew coordinators. It should provide the capability to display and retrieve information, to alert the crew coordinator and monitor the operation, to solve and evaluate different solutions, and to automatically implement the solution.

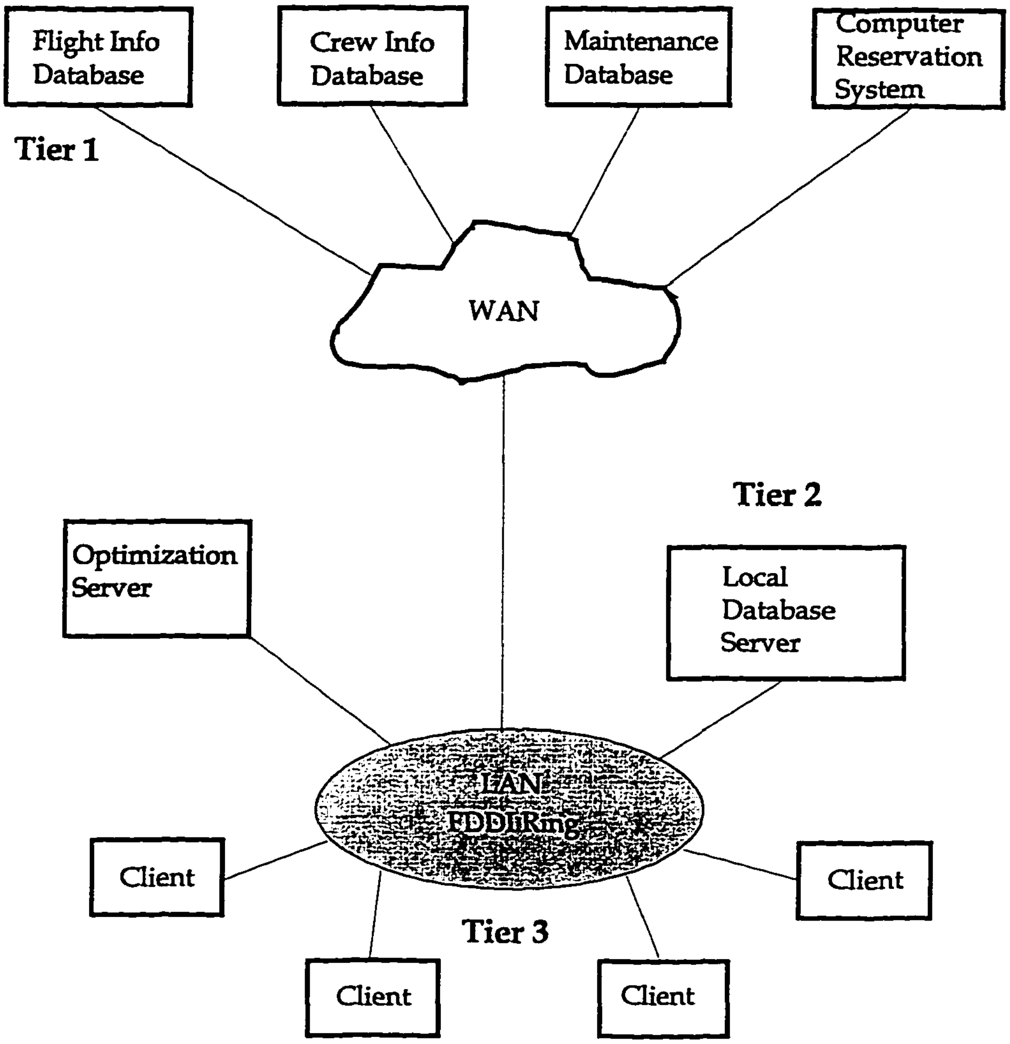

Figure 5.1 The decision support environment for crew management.

Based on the above requirements, we propose a three-tier client/server architecture for the decision support environment (see Figure 5.1). Tier 1 consists of the mainframe databases and application programs, such as flight schedule system, crew databases, maintenance database, reservation system, etc. The second tier is the local database server. This server collects and stores all the relevant information and messages from various mainframe systems for the applications in the crew management environment. The first two tiers communicate via the corporate wide-area network (WAN). Tier 3 is the client applications which usually feature user-friendly graphical user interface (GUI). There is a dedicated application server which hosts the solver program for crew management. It may also contain other applications. Tier 2 and Tier 3 share a FDDI ring local-area network (LAN). The fiber optics LAN is adopted due to the large amount of traffic within the environment and high-speed requirement of the systems.

The heterogeneous environment is tied together by middleware. Middleware is an enabling layer of software which supports multiple communication protocols, multiple programming languages and multiple execution platforms. It resides between the business applications and the network infrastructure of multiple protocols.

Chapter 6 Future Works

In previous chapters, we have discussed airline operation in general and irregular operation in particular. We have developed a robust framework for solving crew management problem during irregular operation and demonstrated that it is efficient enough to be deployed as a real-time decision support system. In this last chapter, we discuss some of the future directions that can be taken to extend and expand our current works. Although many minor improvements can be made to the current algorithm, such as adding more heuristics at different places to improve the efficiency of the algorithm, thus allowing more time to search for better solutions, we believe that at a much higher level, there are two directions in which we can greatly enhance the functionality and usability of the current system: First, the crew operation can be integrated with other aspects of the overall irregular operations; Second, the system can be made more user-friendly and interactive. Here, we do not mean better GUIs or any cosmetic changes, we mean more powerful functionality that will require changes in the algorithm. We elaborate in the following two sections.

6.1 AN MORE INTEGRATIVE IRREGULAR OPERATION MODEL

The crew model for irregular operations as is developed in this dissertation solves only the crew part of an overall irregular operation problem. It assumes that crew problem is relatively independent from other aspects of the irregular operation. This model reflects, in large part, the way major airlines handle irregular operations, i.e., the operations manager solves the aircraft problem first and then crew coordinator solves the crew problem. The problem with the current model is, it ignores the interrelation between different aspects of irregular operation, particularly that between the aircraft rerouting and the crew pairing repair. It assumes that aircraft rerouting has been done and a new, repaired flight schedule has been decided by operations manager. Based on this assumption, it fixes any crew pairings that are broken and crew any flights that may have a shortage of crew under the new flight schedule. In practice, this interrelation is reflected or handled by the close communication between operations managers and crew coordinators, with operations managers driving the whole process and crew coordinators reacting to the changes in flight schedule. The crew model in this dissertation can also be used in a "what-if" mode, i.e., the operations manager make a change in flight schedule and ask the crew coordinator to find crew solution. The crew coordinator in turn runs the system and tells operations manager if the new flight schedule is feasible. This process may be repeated several times until a satisfactory solution to the overall problem is found. The drawbacks of this operation model are that it requires human intervention to facilitate this iterative process, which is rather inefficient and, more important, it treats both the aircraft rerouting problem and crew pairing repair problem as though they were isolated problems.

A more integrative model should combine the different pieces of irregular operation, especially the aircraft rerouting model and crew pairing repair model. The crew part of the irregular operation and the aircraft part are intricately related. For some crew problems, a simple aircraft swap may save the day whereas if it is limited to crew manipulation, the solution may be very complicated or it does not even exist at all.

Recently, some impressive work have been done on the aircraft rerouting during irregular operation (M. Argüello [15]). They have shown that using GRASP-based heuristic algorithm, an entire fleet can be solved rather efficiently. In the future, it is hoped that our algorithm for crew part can be combined with their work to form a more complete and integrative framework.

6.2 A MORE REACTIVE SYSTEM

The system as it is now has the capability to allow user to input various parameters, such as recovery time, solving region (the system can also find a region itself for a particular problem), and/or fixing certain pairings, etc., to adjust for different problems. These definitely have made the system more flexible and friendly. On the other hand, they are still limited in functionality. A more interactive system would not only allow user to set parameters *before* the solving process, it should also permit the user to adjust the obtained solutions *after* the solving process. That is, if user does not like some part of the solution, he/she should be able to fix some variables, change others within the solution and let the system to get a new set of solution to accommodate the new constraints or requirements and, this process should be quick enough and should not treat the problem as a brand new one and embark on another whole solving process. This kind of demand is brought up from the perspective of a user and it can not be met by manipulating user interface. It requires some changes or modifications of the algorithm.

Appendix A Federal Aviation Regulations

This appendix documents the most important Federal Aviation Regulations (FARs). They are not the complete FARs, but the ones that are more important from the algorithm implementation standpoint. Neither does this appendix include the non-FAR legalities, which may include company policies and labor contracts between the airline and the pilots and flight attendants. For each of the rules, either the original text of the FAR rules or a summary description of the corresponding rules is presented. These rules are compiled and organized in such ways that they are easy to understand and implement.

A.1. FAR Domestic 8 in 24 Minimum Rest

This is the text of the FAR rule:

> *FAR 121.471*
>
> *(b.) Except as provided in paragraph (c) of this section, no domestic air carrier may schedule a flight crewmember and no flight crewmember may accept an assignment for flight time during the 24 consecutive hours preceding the schedule completion of any flight segment without a scheduled rest period during that 24 hours of at least the following:*
>
> 1. *9 consecutive hours of rest for less than 8 hours of scheduled flight time.*
> 2. *10 consecutive hours of rest for 8 or more hours of scheduled flight time.*
> 3. *11 consecutive hours of rest for 9 or more hours of scheduled flight time.*

> *An air carrier may schedule a flight crewmember for less than the rest required in paragraph (b) of this section or may reduce a scheduled rest under the following conditions:*
>
> 1. *A rest required under paragraph (b)(1) of this section may be scheduled for or reduced to a minimum of 8 hours if the flight crewmember is given a rest period of at least 10 hours that must begin no later than 24 hours after the commencement of the reduced rest period.*
>
> 2. *A rest required under paragraph (b)(2) of this section may be scheduled for or reduced to a minimum of 8 hours if the flight crewmember is given a rest period of at least 11 hours that must begin no later than 24 hours after the commencement of the reduced rest period.*
>
> 3. *A rest required under paragraph (b)(3) of this section may be scheduled for or reduced to a minimum of 9 hours if the flight crewmember is given a rest period of at least 12 hours that must begin no later than 24 hours after the commencement of the reduced rest period.*

A.2. FAR Minimum Rest and Maximum Duty

This legality stipulates the flight attendants' FAR duty period limitations and rest requirements. This is a summary description of the FAR rule:

> *This rule requires that a flight attendant be given 9 hours of rest following up to 14 hours of scheduled flight. The 9-hour period could be reduced to as little as eight hours, if the employer schedules a 10-hour rest period following the next duty period. Duty periods of 14 hours and greater should be followed by a rest of 12 hours. The 12-hour rest can be reduced to no less than 10 hours, if followed by a 14-hour rest after the next duty period.*
>
> *The domestic maximum duty period is 18 hours. Only on international flight may the duty period be extended to 20 hours.*

A.3. FAR International 2-Man Crew Required Rest

This legality is the FAR 2-crew international flight time limitations and rest requirements. This is the text of the FAR rule:

> *FAR 121.481 Flight time limitations: One or two pilot crews*
>
> *(b.) If a flag air carrier schedules a pilot to fly more than eight hours during any 24 consecutive hours, it shall give him an intervening rest period at or before the end of eight scheduled hours of flight duty. This rest period must be at least twice the number of hours flown since the preceding rest period, but not less than eight hours. The air carrier shall relieve that pilot of all duty with it during that rest period.*
>
> *(c.) Each pilot who has flown more than eight hours during 24 consecutive hours must be given at least 18 hours of rest before being assigned to any duty with the air carrier.*

A.4. FAR 3-Crew International 12 in 24—Maximum Flight Time in 24 Hours

This legality stipulates the FAR 3-crew international flight time limitation of 12 hours of flying in 24 hours. This is the text of the FAR rule:

> *FAR 121.483*
>
> *(a.) No flag air carrier may schedule a pilot to fly, in an airplane that has a crew of two pilots and at least one additional flight crewmember, for a total of more than 12 hours during any 24 consecutive hours.*

A.5. FAR 4-Crew International Double Out Rest

This legality stipulates the FAR 4-crew international rest requirement of double the flight time since the preceding rest at home base. This is the text of the FAR rule:

> *FAR 121.485 Flight time limitations: Three or more pilots and an additional crewmember:*

> *(b.) The flag air carrier shall give each pilot, upon return to his base from any flight or series of flights, a rest period that is at least twice the total number of hours he flew since the last rest period at his base.*

A.6. FAR 3-Crew International 20 in 48 and 24 in 72—Flight Time Limit and Required Rest in 48 Hours and 72 Hours

This legality stipulates the FAR 3-crew international rest requirements for flight time of 20 or more hours of flying in 48 hours and 24 or more hours of flying in 72 hours. This is the text of the FAR rule:

> *FAR 121.483 Flight time limitations: Two pilots and one additional flight crewmember.*
>
> *(b.) If a pilot has flown 20 or more hours during any 48 consecutive hours or 24 or more hours during any 72 consecutive hours, he must be given at least 18 hours of rest before being assigned to any duty with air carrier.*

A.7. FAR Domestic 30 in 7, and 2-pilot International 32 in 7 Flight Time Limitations in 7 Days

This legality stipulates the FAR domestic flight time limitation of 30 hours in 7 days, and the international flight time limitation of 32 hours in 7 days. This is the text of the FAR rules:

> *FAR 121.471 (Domestic)*
>
> *(a.) No domestic air carrier may schedule any flight crewmember and no flight crewmember may accept an assignment for flight time in scheduled air transportation or in other commercial flying if that crewmember's total flight time in all commercial flying will exceed—*
>
> *(3) 30 hours in any 7 consecutive days.*
>
> *FAR 121.481 (International) Flight time limitations: One or two pilot crews.*
>
> *(d.) No pilot may fly more than 32 hours during any seven consecutive days...*

A.8. FAR Domestic and 2-Pilot and 3-Pilot International 1 in 7

This legality stipulates the FAR domestic and international rules of a minimum of 24 hours free from duty in any 7 consecutive days. This is the text of the FAR rules:

> *FAR 121.471 (Domestic)*
>
> *(d.) Each domestic air carrier shall relieve each flight crewmember engaged in scheduled air transportation from all further duty for at least 24 consecutive hours during any 7 consecutive days.*
>
> *FAR 121.481 (International) Flight time limitations: One or two pilot crews.*
>
> *(d.) ...and each pilot must be relieved from all duty for at least 24 consecutive hours at least once during any seven consecutive days.*
>
> *FAR 121.483 (International) Flight time limitations: Two pilots and one additional flight crewmember.*
>
> *(b.) ...In any case he must be given at least 24 consecutive hours of rest during any seven consecutive days.*

A.9. FAR Domestic and 2-Pilot International Max Flight Time per Month and 3-Pilot International Max Flight Time in 30 Days

This legality stipulates the FAR domestic and 2-pilot international 100 hours maximum flight time per calendar month, and 3-pilot international maximum flight time in 30 days. This is the text of the FAR rules:

> *FAR 121.471 (Domestic)*
>
> *(a.) No domestic air carrier may schedule any flight crewmember and no flight crewmember may accept an assignment for flight time in scheduled air transportation or in other commercial flying if that crewmember's total flight time in all commercial flying will exceed—*
>
> *(2) 100 hours in any calendar month.*

FAR 121.481 (International) Flight time limitations: One or two pilot crews.

(e.) No pilot may fly as a member of a crew more than 100 hours during any one calendar month.

FAR 121.483 (International) Flight time limitations: Two pilots and one additional flight crewmember.

(c.) No pilot may fly as a flight crewmember more than—

(1) 120 hours during any 30 consecutive days.

A.10. FAR Domestic and 2-Man Crew International Flight Time per Duty Period

This legality stipulates the FAR 2-crew domestic and international flight time per duty period limitations. This is the text of the FAR rule:

FAR 121.471

(a.) No domestic air carrier may schedule any flight crewmember and no flight crewmember may accept an assignment for flight time in scheduled air transportation or in other commercial fling if that crewmember's total flight time in all commercial flying exceed—

(4) 8 hours between required rest periods.

FAR 121.481 Flight time limitations: One or two pilot crews.

(a.) A flag air carrier may schedule a pilot to fly in an airplane that has a crew of one or two pilots for eight hours or less during any 24 consecutive hours without a rest period during these eight hours.

Appendix B Data Models

In this appendix, the data models used in this dissertation for crew pairing repair problem is defined and discussed. The algorithms and data models developed in this dissertation are the most important building blocks of a large-scale decision support system for a major North America airline. Compared with the data models used in the real production system, the data models presented in this appendix are simplified so that the most essential aspects of the problem are adequately reflected, yet many non-essential details are ignored.

The main components of the data models are the flight schedule table, the crew pairing table, the crew assignment table, the flight leg table, and some parameter tables that mostly have to do with legalities. The first four tables are discussed in this appendix since they are the most important ones.

DATA SCOPE

An operating window must be determined before any data can be provided. The length of the window is determined by the number of days it takes for the airline to recover its crew schedule after irregular operations in the worst case. Obviously, this length can be somewhat arbitrary, since it is hard to predict the number of days it takes the airline to recover their system in any particular irregular operation. In practice, though, 5-day is considered an appropriate length of window in which almost all crew pairing repair problems can be solved. This is also the length which is adopted in this dissertation. In other words, the flight schedule table consists of all the flights between today and today+4. Since crew pairings usually span 2 to 4 days, with even longer pairings possible, any pairing which touches the above operating window is included in the crew pairing table, even if portion of the pairing is outside the window. Likewise, crew assignment table consists of all the duty assignments (including pairings, reserve assignments and off time) that are inside the window entirely or touch the window partially. This inclusion of all pairings and assignments that touch the window is to facilitate the legality checking purpose.

FLIGHT SCHEDULE TABLE

Table B-1 Flight Schedule Table

Field				Data Type
No.	**Description**	**Key**	**Length**	
1	Flight Number	X	4	Alphanumeric
2	Origination City	X	3	Alphanumeric
3	Destination City		3	Alphanumeric
4	Equipment		3	Alphanumeric
5	Departure Date (YYYYMMDD)	X	8	Alphanumeric
6	Departure Time (GMT)	X	4	Alphanumeric
7	Arrival Date (YYYYMMDD)		8	Alphanumeric
8	Arrival Time (GMT)		4	Alphanumeric
9	Flight Time		4	Alphanumeric
10	Flight Status		2	Alphanumeric
11	Aircraft Tailnum		4	Alphanumeric

Table B-1 lists the data format for the flight schedule table. Note that the Greenwich Mean Time (GMT) is used throughout for all the time fields. Flight schedule table contains every flight that is within the predetermined operating window.

CREW PAIRING TABLE

Table B-2 Crew Pairing Table

Field				Data Type
No.	Description	Key	Length	
1	Pairing Class	X	1	Alphanumeric
2	Pairing Number	X	6	Alphanumeric
3	Pairing Date	X	8	Alphanumeric
4	Effective From Date		8	Alphanumeric
5	Effective To Date		8	Alphanumeric
6	Frequency		4	Alphanumeric
7	Following occurs Number of Duty-Periods Times			
7a	Following occurs Number of Flight Legs Times			
7a-1	Flight Date (YYYYMMDD)		8	Alphanumeric
7a-2	Flight Number		4	Alphanumeric
7a-3	Origination City		3	Alphanumeric
7a-4	Destination City		3	Alphanumeric
7a-5	Depature Time (GMT)		4	Alphanumeric
7a-6	Arrival Time (GMT)			Alphanumeric
7a-7	Equipment		3	Alphanumeric
7a-8	Flight Leg Status		2	Alphanumeric
7a-9	Deadhead Indicator		1	Alphanumeric
7a-10	Aircraft Tailnum		4	Alphanumeric
7b	Rest Type		2	Alphanumeric
7c	Duty Period Number		2	Numeric
7d	Brief Time		4	Alphanumeric
7e	Debrief Time		4	Alphanumeric
7f	Accumulated Flight Time		4	Alphanumeric
7g	Accumulated Duty Time		4	Alphanumeric
7h	Rest Time		4	Alphanumeric

A crew pairing consists of several duty period, which, in turn, is made up of a sequence of flight legs followed by a duty break or rest period. Table B-2 shows the data format for the pairing table.

CREW TABLE

Crew table contains all the assignments for every crew member of the airline. These assignments include pairing assignments, reserve duty assignment and off time. Only assignments that are inside or overlap with the operating window are included, however. Table B-3 gives the data format for the crew table.

Table B-3 Crew Table

Field				Data Type
No.	**Description**	**Key**	**Length**	
1	Last Name		10	Alphanumeric
2	First Name		10	Alphanumeric
3	Employee Number	X	10	Alphanumeric
4	Qualification		2	Alphanumeric
5	Base		3	Alphanumeric
6	Equipment		3	Alphanumeric
7	Flowing occurs Number of Block-Time's Times			
7a	Pairing or Assignment		6	Alphanumeric
7b	Class ('P'ilot or 'F'light Attendant)		1	Alphanumeric
7c	Block Start Date (YYYYMMDD)		8	Alphanumeric
7d	Block Start Time (GMT)		4	Alphanumeric
7e	Block End Date (YYYYMMDD)		8	Alphanumeric
7f	Block End Time (GMT)		4	Alphanumeric
7g	Position		4	Alphanumeric
7h	Line Type ('L'ine-holder or 'R'eserve)		1	Alphanumeric

FLIGHT LEG TABLE

Table B-4 is the data format for Flight Leg Table. A flight leg record follows each flight schedule record. The flight leg record contains the information for all the crew member on the flight preceding it.

Table B-4 Flight Leg Table (Attached to Flight Schedule Table)

Field				**Data Type**
No.	**Description**	**Key**	**Length**	
1	Following occurs Number of Crew Members Times			
1-a	Class		1	Alphanumeric
1-b	Pairing Number		6	Alphanumeric
1-c	Flight Date (YYYYMMDD)		8	Alphanumeric
1-d	Position		4	Alphanumeric
1-e	Employee Number		10	Alphanumeric

Glossary

Base
A geographical location where pilots and flight attendants are stationed

Block to Block
The period of time beginning when an aircraft first moves from the blocks for the purpose of flight, and ending when aircraft comes to a stop at the blocks at the next point of landing, or at the point of departure if the flight returns without becoming airborne.

Calendar Day
Midnight to midnight local domicile time (LDT)

Captain
A pilot who is in command of the aircraft and has authority over all cabin and cockpit personnel while on flight duty.

Deadhead
Crewmember(s) flying to or from a flight or operational assignment.

Debrief
The time allowed for completion of post flight duties. Debrief ends at Block-in plus 15 minutes except, Block-in plus 30 minutes when Customs clearance is required.

Domicile
A specific geographic location which is designated by the airline as a Base.

Duty Period
The elapsed time from the time a pilot is required to report for duty (or deadheading to or from duty) or the actual reporting time, whichever is later, until the time the pilot is released from duty after block-in of the last flight segment flown or deadheaded before a minimum rest period of day off.

First Officer
A pilot who is second in command of the aircraft.

Flight Time
The first movement of an aircraft for the purpose of flight until it comes to rest at the next point of landing. (See Block to Block)

International Relief Officer
A pilot qualified to fly as a Supplemental Crewmember on international flights.

Pairing
A series of flight segments (a flight between two city pairs). Pairings may include deadhead.

Reassignment
A change to a crewmember's schedule as a result of an operational contingency.

Rest Period
The period of time between the end of a debrief and the report time of the next trip.

Second Officer
A pilot who is third in command of the aircraft.

Bibliography

[1] Subramania, R., R. P. Scheff, J. D. Quillinian, D. S. Wiper and R. E. Marsten. 1994. Coldstart: Fleet Assignment at Delta Air Lines. *Interfaces* 24, 104-120.

[2] Gershkoff, Ira. 1989. Optimizing Flight Crew Schedules. *Interfaces* 19, 29-43.

[3] Anbill, R., E. Gelman, B. Patty and R. Tanga. 1991. Recent Advances in Crew-Pairings Optimization at American Airlines. *Interfaces* 21, 62-74.

[4] Jarrah, A., G. Yu, N. Krishnamurthy and A. Rakshit. 1993. A Decision Support Framework for Airline Flight Cancellations and Delays. *Transportation Science*, Vol. 27, No 3. 266-280.

[5] McCartney, S. Nov. 12, 1995. Driver's Seat: Why Are Big Airlines Considering Mergers? *Wall Street Journal*. A1.

[6] Garey, M and D. Johnson. 1979. Computers and Intractability—A Guide to the Theory of NP-Completeness. W. H. Freeman and Co. 221.

[7] Anbil, R., R. Tanga and E. L. Johnson. 1992. A Global Approach to Crew-Pairing Optimization. *IBM System Journals*, Vol 31, No 1. 71-78.

[8] Baues, G., P. Kay and P. Charlier. Constraint-based Resource Allocation for Airline Crew Management. *Internal Publication*, Transport Application Group, COSYTEC SA.

[9] Luo, S. and G. Yu. Airline Schedule Perturbation Problem Caused by the FAA's Group Delay Program. *Transportation Science*, forthcoming.

[10] Yu, G. 1995. An Optimization Model for Airlines' Irregular Operations Control. *Proceedings of the International Symposium on Operations Research with Applications in Engineering, Technology, and Management*. Beijing, China.

[11] Mathaisel, D.F.X. 1996. Decision Support for Airline System Operations Control and Irregular Operations. *Computers and Operations Research*, 23, 11, 1083-1098.

[12] Press, H. W., et al. 1992. Numerical Recipes in C, 2nd ed. Cambridge University Press, New York.

[13] Pearl, J. 1985. Heuristics—Intelligent Search Strategies for Computer Problem Solving. Addison Wesley, New York. 33-39.

[14] Ahuja, R., T. Magnanti and J. Orlin. 1993. Network Flow, Theory, Algorithms, and Applications. Prentice Hall. 107-108.

[15] Michael Argüello. 1997. Framework for Exact Solutions and Heuristics for Approximate Solutions to Airlines' Irregular Operations Control Aircraft Routing Problem. Ph.D. dissertation, Department of Mechanical Engineering, University of Texas at Austin.

[16] Kernighan, B. W., and S. Lin. 1970. An Efficient Heuristic Procedure for Partitioning Graphs, *Bell System Tech. J.*, 49, No. 2, 291-307.

[17] Lin, S. 1965. Computer Solutions of the Traveling Salesman Problem, *Bell System Tech. J.*, 44, No. 10, 2245-69.

Vita

Guo Wei was born in Xi'an, Shaanxi Province of China on June 29, 1964, the eldest son of Wei Pingde and Jia Liquan. After completing his work at No.6 High School, Xian, Shaanxi Province of China, in 1982, he entered Beijing University in Beijing, China to study physics. He received the degree of Bachelor of Science from Beijing University in July 1986. Between September 1986 and June 1989, he continued his study and research in physics at the Institute of Physics, Chinese Academy of Sciences in Beijing, China. In August 1989, he came to the U.S. to pursue a doctoral degree in physics at the University of Texas at Austin. Almost one year after he passed his Ph.D. qualification exam, he decided business was more interesting and suitable for him. In September 1992, he became a doctoral student at the Graduate School of Business, the University of Texas at Austin, majoring in Management Science. During the following years he had been a consultant to airline industry and helped start a consulting company in Operations Research and Management Science in Austin, Texas.

Permanent address: 3600 Greystone #822, Austin, Texas 78731

This dissertation was typed by the author.