Integrated Airline Schedule Optimization: Models and Solution Methods

by

Rajesh Gopalakrishna Shenoi

Bachelor of Technology in Civil Engineering
Indian Institute of Technology, Madras, India (1991)
Master of Science in Transportation
Massachusetts Institute of Technology (1994)

Submitted to the Department of Civil and Environmental Engineering
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in Transportation Systems and Decision Sciences

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 1996

Author .....
Department of Civil and Environmental Engineering
June 28th, 1996

Certified by.. .....
Cynthia Barnhart
Associate Professor
Thesis Supervisor

Accepted by . .....
Joseph Sussman
Chairman, Departmental Committee on Graduate Students

Integrated Airline Schedule Optimization: Models and Solution Methods

by

Rajesh Gopalakrishna Shenoi

Submitted to the Department of Civil and Environmental Engineering
on June 28th, 1996, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
in Transportation Systems and Decision Sciences

Abstract

The airline schedule planning problem is defined as the sequence of decisions that need to be made to make a flight schedule operational. Given the high level of competition in the airline industry, effective decision making is crucial to the profitability of an airline. This is the motivation for this dissertation in which we focus on long-haul airline schedule planning problems and attempt to integrate the decision making process. Our goal is to achieve simultaneous rather than sequential solution, because a simultaneous solution will generate more economical solutions and create fewer incompatibilities between the decisions. Specifically, we develop integrated models called the *aircraft scheduling* model, and the *crew scheduling* model.

The objective of the aircraft scheduling model is to combine *fleet assignment*, i.e., the problem of assigning aircraft types to flights, *through flight assignment*, i.e., the problem of assigning aircraft to *sequences* of flights for revenue considerations, and *aircraft maintenance routing*, i.e., the problem of assigning aircraft to flights so maintenance requirements are satisfied. We show that sequential solution of these modules does not always yield feasible solutions, and this motivates the need for the integrated aircraft scheduling model. We develop specialized solution methods to solve this problem, and using data from a long-haul carrier, we obtain near-optimal solutions to the integrated model.

The crew scheduling models we develop attempt to model both the *crew pairing* and *deadhead selection* problems. The long-haul crew pairing problem requires the coverage of a set of long-haul flights by a minimum cost set of crew schedules. We present an alternate model and solution approach for this problem, and show how the alternate model can also be used to select deadheads, i.e., flights that do not have to be flown by the airline but help to reposition the crew to improve their productivity. Using our approach, computational experiments with data provided by a long-haul airline show that significantly improved solutions can be achieved quickly. Compared to existing methods, solution costs are reduced by about 20% (translating into annual cost savings of millions of dollars) and run times are reduced by about 80%.

We then attempt to integrate the aircraft and crew scheduling problems. Since these two decisions have a huge impact on costs, simultaneous, rather than sequential, solution could have a tremendous impact on the bottom line. Given that the state-of-the-art is still far from solving such a model, we combine approximate models of the crew pairing and fleet assignment problems. These approximate *surrogate* models are potentially tractable and good representations of the problem, and can be used as a first step in generating integrated solutions to the aircraft and crew scheduling problems. We finally present directions for future research.

Thesis Supervisor: Cynthia Barnhart
Title: Associate Professor

Acknowledgments

I would first like to thank my advisor, Prof. Cynthia Barnhart, for being a constant source of inspiration, a role model, a mentor, and a friend. I would also like to thank my committee members, Prof. Ellis L. Johnson and Prof. Yosef Sheffi, for their invaluable guidance.

I would like to thank all the faculty at the Center for Transportation Studies for giving me a great education. I would like to thank Prof. Pam Vance of Auburn University, and Prof. George Nemhauser, Prof. Lloyd Clarke, and Prof. Martin Savelsbergh of Georgia Tech, who were a constant source of help for all these years. I would like to thank the employee owners of United Airlines, Nirup Krishnamurthy, Ahmad Jarrah, Alamuru Krishna, Deepa Mahidhara, Venkatram Krishnamurthy, and Muthukrishnan for all the technical support they have rendered.

I would like to thank Maria Marangiello, Lisa Magnano Bleheen, and Nancy Martin at the Center for Transportation Studies, and Cynthia Stewart, Jessie Williamson, and Pat Dixon at the Civil Engineering headquarters, for helping me out with various administrative details.

I would like to thank Chris Caplice, Daeki Kim, Bill Cowart, Rajiv Lochan, Daniel Rodriquez, Owen Chen, Oh Kyoung Kwon, Dan Turk, and Hans Klein for being wonderful officemates. I also must thank all the members of LSOG, Brian, Paul, Gendron, Bill, Jung, Andras, Tim, Jim, and Fransisco for those great discussions.

I would like to thank my friends, Sudhir, Hari, Dinesh, Prodyutt, Denise, Harshdeep, Gokul, Salal, Sankar, Srikar, Ashok, Pat, Nabha, Kishore, Ninja, Kevin, Gorti, Jaime, Pradeep, Sanjeev, Sharath, Amit, Rahul, Srikant, Manish (all of them), Kazi, Milly, Cathal, Ayelet, Tim, Antulio, John, Qi, Raju, Cartikeya, Bhavin, Atul, Ravi, Mantripragada, Naren, Barua, Nagi, Anil, Adriana, Amjad, and Rabi, for all the fun times at MIT.

This would not be complete without the mention of the most important people in my life. I thank my family for showering me with love and affection, for constantly believing in me, and for always being a source of strength. I would like to thank my fiancee's family for their support, and for motivating me to study a new language. I would also like to thank my fiancee, Ana Laplaza, without whose support, friendship, and love, this dissertation would not be complete.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Since deregulation, the U.S. airline industry has been extremely competitive. In order to gain a competitive edge, airlines are developing and investing in decision support tools to help them lower operating costs and increase revenues and profits. This dissertation focusses on the airline scheduling process and presents mathematical models and solution methods for selected scheduling problems.

1.1 Profitability of the Airline Industry

Since deregulation, in 1978, fierce competition in the airline industry has decreased its profitability. While the industry was generally profitable before deregulation, it has reported net losses in nine of the seventeen years following deregulation (Figure 1-1[1]). Between 1990 and 1994 alone, the ten largest American carriers have posted cumulative losses of $13 billion [ECON 1996][2]. A *major carrier* as defined by the Department of Transportation (DOT) is an airline with more than a billion dollars in annual revenues [Nugent 1993]. We refer to all other airlines as *small* carriers.

The various reasons suggested for this include high fuel and labor costs, slowing traffic growth, arcane bankruptcy laws [ECON 1995c, 1996], and even high federal taxes [Dasburg 1995]. In order to analyze an industry's profitability, Porter [1980]

[1]Source: World Wide Web URL: http://www.air-transport.org/data/earnings.htm
[2]ECON refers to the *The Economist* and WSJ refers to *The Wall Street Journal.*

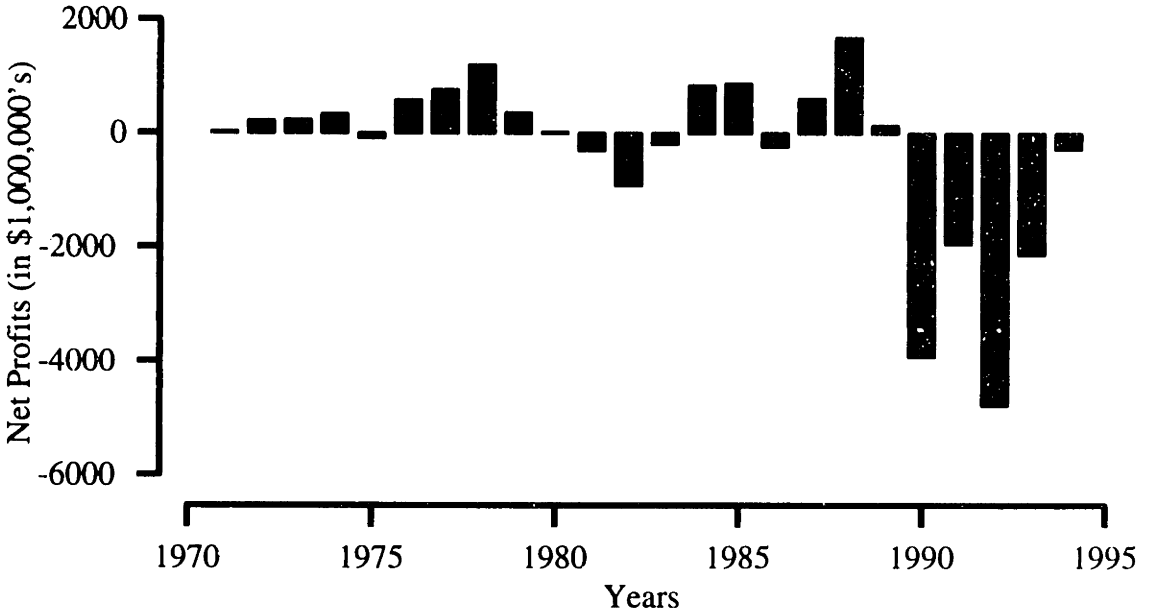

Figure 1-1: Airline Historical Net Profits (1971-1994)

suggests the analysis of five key factors:

1. Threat of entry;
2. Competition between existing companies;
3. Competition from substitutes;
4. Market power of buyers; and
5. Market power of suppliers.

We analyze the effect of each of these factors on the profitability and competition in the airline industry.

1.1.1 Threat of entry

Barriers to entry into the U.S. airline industry are currently at one of their lowest levels [McCartney 1996b]. In the last three years alone, the U.S. Department of Transportation (DOT) has certified upto thirty new carriers. Barriers to entry have fallen for a variety of reasons.

Diseconomies of Scale

Airlines are believed to possess little or no economies of scale for various reasons [Shepherd and Brock 1995]. Larger airlines tend to have unionized labor, while smaller airlines do not. Low-fare startups pay their pilots about half the salary offered by established airlines [ECON 1996]. Delta is starting a new *Lite* service for which a captain with 8 years experience gets paid $108,000 as compared to $150,000 on normal routes [Brannigan and De Lisser 1996]. In addition, small airlines have lower costs by providing minimal service, such as ticketless travel [McCartney 1996b]. Big airlines also have the cost disadvantage that is concomitant with building and operating expensive hubs [Shepherd and Brock 1995].

Changing Technology

Companies, like World Technology systems, have promoted the idea of hiring out important business functions. World Technology Systems handles, for airlines, everything from baggage handling, ticket reservations, to finding aircraft and the crew to fly them. Using their services, Sterling One, a California firm, was able to offer a scheduled flight from Long Beach, California to Chicago, with just five employees and three months of preparation [McCartney 1996b].

Government Support

In order to encourage new carriers, the U.S. Department of Transportation has set up a new office to handle applications [McCartney 1996b]. In addition, each applicant is being charged only $1,000, a subsidy over the $150,000 that it costs the DOT to process the application.

1.1.2 Competition between Existing Companies

The feverish competition among U.S. carriers culminated in 1992, when AMR Corporation's *value pricing* scheme [Michael and Silk 1994, 1994b] sparked off a fare war that cost the industry a $3 billion loss that year [Nugent 1993]. While the situa-

tion has stabilised (the airlines managed a profit of $2 billion in 1995 [ECON 1996]), margins have been severely eroded for reasons discussed below.

Low-Fare Carriers

The success of Southwest has prompted a new wave of low-fare carriers [ECON 1996], which in addition to lowering profit margins and increasing competition, have already cornered 15% of the domestic market [ECON 1995c], and are currently flying on routes that carry 37% of domestic airline traffic [Gruley and McCartney 1995]. American airlines [McCartney 1996b] and Delta [Brannigan and De Lisser 1996] now have low-fare competitors on 60% of their domestic routes. The U.S. DOT estimates that these new airlines are saving consumers about $4 billion annually [Gruley and McCartney 1995].

Low-fare carriers have had a huge impact on *fortress-hubs*, where the dominant airline supposedly blocks out competition and charges a *hub-premium* [McCarthy 1996]. United and Delta have a 70% and 72% market share at Denver and Atlanta respectively. Low-fare carriers are cutting into the premiums of these big airlines at their respective hubs [McCarthy 1996], and are also targeting high margin markets [Jaffe 1996, Brannigan 1996].

The government has played a part in forging airline competition. The assault on the fortress hubs by the small entrants was helped by the actions of the Clinton Administration which pushed a few key anti-trust cases that protected small carriers [McCarthy 1996]. As another example, the government has been investigating complaints by small carriers about possible anti-competitive moves on the part of the majors [Nomani 1996].

Price Competition

Airlines constantly compete on price. A few months ago, Delta cut fares in 16,000 markets by uptc 40% for travel in spring [WSJ 1996e]. This was soon followed by a summer fare cut by four other major carriers [WSJ 1996c], and Southwest [WSJ 1996]. Similar discounts have been offered in earlier years [Patterson 1995, Quintanilla

1995]. They are now becoming so common that leisure travellers seem to expect a discount as an incentive to travel [McCartney 1996e].

Brand Loyalty

Airlines also compete with each other to win consumers' brand loyalty, with the use of *Frequent-Flyer* programs [McCartney 1996]. While these help to keep customers loyal, it can sometimes work against U.S. carriers. A case in point is American Airlines South American Operations. Analysts believe that the AMR corporation has made cumulative operating profits exceeding $500 million from its South American routes since 1990, and American carriers have more than a 50% market share of all traffic between the U.S. and South America. In order to win back market share and regain profitability, 14 South American airlines have jointly created *Latin Pass* - a scheme that lets travellers pool the miles earned from any of the member airlines [ECON 1995e].

"Open Skies"

An *open skies* agreement between two countries removes all restrictions on air travel between those two countries. Open skies agreements have been signed by the U.S. with countries such as Holland, Canada, and Germany [ECON 1995c, Carey 1996b]. Such agreements boost competition and decrease margins due to increases in capacity. For example, in the second half of 1995, passenger traffic between the U.S. and Canada rose 22% while the number of flights rose 43% [Nomani 1996b].

1.1.3 Competition from Substitutes

The pressure on companies to reduce travel costs has increased in recent years by inexpensive communication technology. Many firms prefer to purchase videoconferencing systems by weighing the cost of such technology against the potential savings they would generate in travel costs [Boroughs 1995]. Telecommunication costs are on the decline due to factors such as the recent telecom deregulation, and competition

from sources such as the internet [ECON 1995, 1995f]. Analysts estimate that deregulation will help drive down telecom prices by upto 70% in the next decade [ECON 1996b]. Today, we have more advanced and cheaper communication equipment than we did a few years ago [Carlton 1995]. PC-based videoconferencing systems are currently available for $1500, and for less than $200 for windows based PC's equipped with multimedia [Bulkeley 1996]. These factors apply a downward pressure on airline demand and revenues.

1.1.4 Market Power of Buyers

Buyer power can come from either agencies representing travellers, or the channels of ticket distribution, i.e., the computer reservation systems.

Buying Power of Travellers

Airline demand is composed of two types of passengers - business travellers, who are insensitive to price, and leisure travellers, who are generally highly price elastic [Shepherd and Brock 1995]. Traditionally, airlines have competed for the business segment because an average business-class fare is about three times an average coach-class fare [ECON 1996]. However, the proportion of leisure travellers is increasing (Table 1.1[3]). It is suggested that leisure travel may account for upto 80% of total air traffic by the year 2000 [Shepherd and Brock 1995].

To make matters worse, airlines may face pressure from decreasing business travel. Cutbacks in airline capacity have raised fares [Brooks 1995], and American companies are estimated to spend $7 billion more on air travel in 1996 than in 1995 [Miller 1995]. This will lead many firms to impose more stringent cost cutting travel restrictions. This is bound to squeeze airline profits, especially in periods of low growth, like the third quarter of 1995 (usually an important quarter) [McCarthy 1995].

Another big problem for airlines is increased buyer power. A company called

[3]Cited in Michael and Silk [1994]. Source: Air Travel Survey, Air Transport Association and Gallup Organisation, reported in *Air Transport 1992*, The Annual Report of the U.S. Scheduled Airline Industry, Washington, D.C.: Air Transport Society of America, June, 1992, p. 10.

Year	Percentage of Adults on	
	Business Travel	Leisure Travel
1981	52%	48%
1989	51%	49%
1990	48%	52%
1991	46%	54%

Table 1.1: Air Travel Mix since 1981

Business Travellers Contractors Corp.(BTCC) is planning to negotiate with airlines, on behalf of corporate clients, for a simpler and less expensive fare structure [Flint 1995, 1995b]. While BTCC promises cost savings due to the simplification of ticket distribution and marketing, and elimination of the middleman, their increased buyer power can help them reduce airline margins.

Buying Power of Ticket Distribution Systems

Tickets are typically purchased by passengers, through travel agents, who use Computerized Reservation Systems (CRS) [Shepherd and Brock 1995]. There is a heavy concentration among CRS's, with the top four systems controlling 83% of the worldwide market. Moreover, there are rumors of a possible merger (between Galileo and Worldspan, two of the big four CRS's) that will create the world's largest CRS [McCarthy and De Lisser 1996]. A greater market concentration among the CRS's can lead to lesser competition between them, and since they constitute the primary ticket distribution channel, this may affect airline margins.

Travel agents can play an important role in airline performance. In 1991, 85% of all airline reservations was done through travel agents [Michael and Silk 1994]. Last year, in order to scale back booking costs, airlines placed a cap on the commissions made by travel agents. Travel agents are now fighting back by suggesting low-fare tricks to passengers such as *back-to-back ticketing*[4], or suggesting low-fare carriers as

[4]Back-to-back ticketing refers to a method of buying two round trip tickets whose itineraries are close enough to swap segments to create virtually two new itineraries. The advantage of doing this is that two round-trip tickets with restrictions can sometimes be less expensive than one unrestricted round-trip ticket [Michael and Silk 1994].

alternatives to the big airlines [Miller 1995b]. The end result will be to reduce airline revenues.

1.1.5 Market Power of Suppliers

The purchases of aircrafts and aviation fuel account for two of the biggest costs for airlines.

Aircraft Manufacturers

Aircraft manufacturing is a capital and technologically intensive industry. To illustrate, the list price of new Airbus A330-200 or a Boeing 767-300ER is about a $100 million [Cole 1996]. It is an extremely concentrated industry and the top three manufacturers - Boeing, Europe's Airbus Industry, and McDonnell Douglas - have achieved a market share of about 94% in the world civilian jet market in the last few years [ECON 1995d]. Any industry with few sellers and many buyers enjoys high profit margins. For example, Boeing is virtually a monopoly in the top end of the market, and some analysts estimate that Boeing makes a profit of about $50 million on every 747 that it sells at $150 million [ECON 1995b]. Moreover, there are talks of a merger between McDonnell Douglas and Boeing [ECON 1995d]. Such a merger would only help to stifle competition and raise prices. Airline suppliers therefore possess market power and the ability to affect airline profits.

Fuel Suppliers

Despite its diminishing share of operating costs (Table 1.2[5]), fuel is still one of the largest expenses in airline operations. In 1993, U.S. carriers spent nearly $10 billion on gasoline, and consumed about 16 billion gallons[6]. Despite this, airline fuel consumption was merely 6.4% of overall U.S. gasoline consumption in 1993[7]. This implies that airlines, individually or collectively, have little market power over oil suppliers.

[5]Source: Statistical Abstracts of the United States: 1995
[6]Source: Fuel Cost and Consumption Tables in *Air Transport World*, 1993-1996
[7]Source: World Almanac and Book of Facts, 1995

Year	Share of Operating cost
1980	29.7%
1985	22.3%
1990	17.6%
1993	12.3%

Table 1.2: Share of Fuel in Operating Costs

This can have disastrous implications for airline profits during periods of rising fuel prices [Sullivan 1996, Page 1996]. For example, a 13 cent increase in aviation fuel prices would have been enough to wipe out the net profits of the airline industry in 1995. In fact, in the first quarter of 1996, the profits of some airlines were hit by rising fuel prices [McCarthy and McCartney 1996].

1.2 Planning in the Airline Industry

The low level of profitability in the airline industry makes decision making crucial. Decisions in this industry are made at various levels that differ in their scope and impact, the amount of information available while making the decision, and the sophistication of the decision making process. We consider two broad classes of decisions - tactical and strategic. Tactical decisions, which we refer to as *schedule planning decisions*, are short term in nature (in the order of months), while strategic decisions have a longer horizon of a few years [Banfe 1992].

1.2.1 Strategic Planning

Strategic decisions typically have long planning horizons of three to five years [Banfe 1992]. Some examples of strategic decisions are given below.

1. A *strategic alliance* is an arrangement where two or more firms combine their strengths to achieve a common goal, such as increasing revenues [Oster 1994]. An example of this in the airline industry is the recent alliance between British Airways and American Airlines [Nomani and Carey 1996]. Such an alliance

would include features such as *code sharing*, a mechanism that allows one airline to channel its passengers into the network of the other, and revenue sharing on certain routes [McCartney 1996d].

2. A major decision for any business is whether to carry out many of its major functions on its own, or contract them out to other firms. The latter practice, called *outsourcing*, has gained popularity especially since young companies may not have either the expertise or the capital to carry out the task themselves. Recently, ValuJet has been considering letting more experienced airlines carry out its maintenance checks [De Lisser and Nomani 1996].

3. The location of the base for its pilots or flight attendants is important, especially for airlines with large international operations. Recently, United airlines has been planning to open bases in Germany and Japan to better serve their international travellers [WSJ 1996d].

4. Strategic planning in an airline also includes bargaining with the labor unions to negotiate wages, increase productivity, and lay down work rules for the crew [Carey 1996].

1.2.2 Schedule Planning

The flight schedule is one of the primary products of an airline [Wells 1994]. While strategic decisions have a big impact on the bottom line, there is enormous scope for an airline to improve its efficiency by better schedule planning. Delta has managed to increase revenues by upto $100 million annually by more efficient assignment of aircraft to flights [Subramanian et al 1994]. American Airlines has shown potential savings of $454 million over the active life of its widebody aircraft by better planning of their heavy overhaul maintenance schedules [Gray 1992]. Millions have been saved annually by airlines through increased crew productivity [Barnhart, Hatay and Johnson 1995]. Thus, given a specific strategic position, an airline can save money or generate greater revenues by better schedule planning.

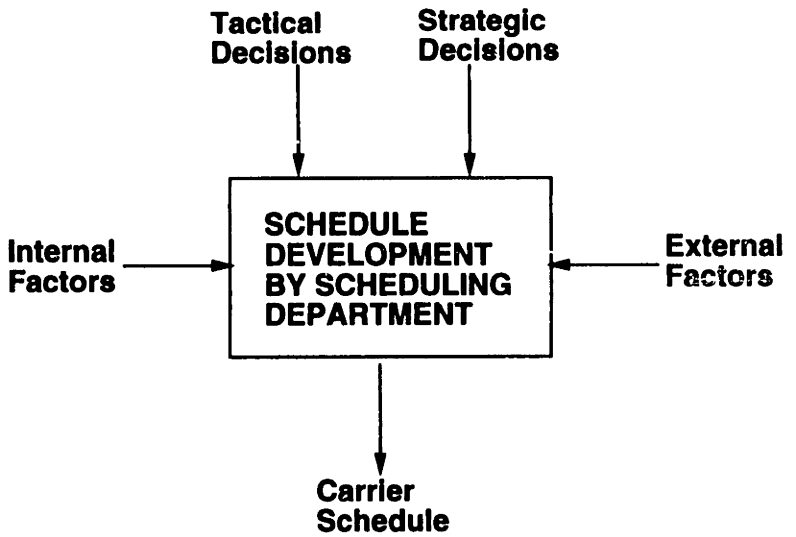

Figure 1-2: The Airline Schedule Development Process

Figure 1-2 shows a framework [Wells 1994] to create an airline schedule. While the development of the schedule is done by the scheduling department, it must consider a variety of *internal* and *external* factors while creating the schedule. Internal factors, i.e., factors that are within the control of the airline, include:

- Crew availability and safety restrictions placed on their flying.
- Availability of aircraft, their fuel capacity and age, noise restrictions at certain airports, or restrictions due to length of runways, etc.
- Availability of maintenance opportunities for these aircraft.
- Availability of facilities such as gates, landing and departure slots, ticket counters, ground equipment and crew, etc.

External factors, i.e., factors outside the airline's control, include:

- Marketing considerations, such as the demand for a specific origin-destination market, which can vary by time-of-day, day-of-week, or by season, or preferences for non-stops vs. single-stops etc.
- Restrictions by airport authorities to reduce peak hour congestion.

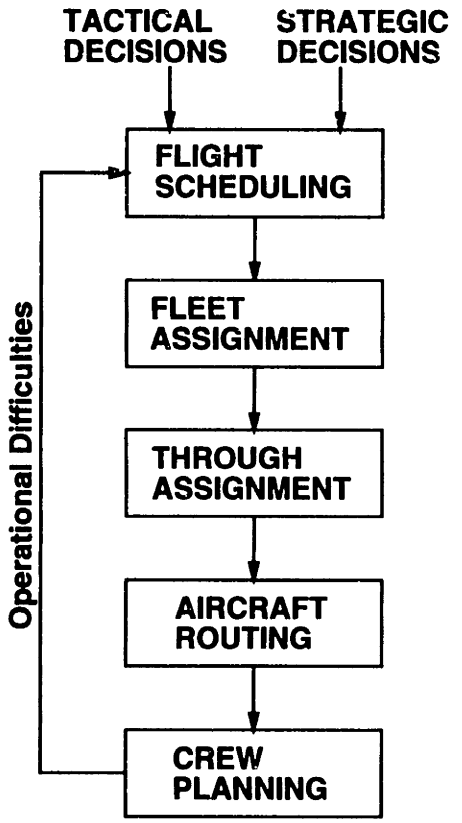

Figure 1-3: A Simplified Scheduling Paradigm

- Noise restrictions by local communities denying flight activity at certain hours.
- Marketing considerations of freight shippers that use the airline.

In addition to these factors, the scheduling department must consider *tactical* and *strategic* decisions. A tactical decision may involve reacting to a rival airline's competing schedule which may severely affect demand forecasts. As examples of strategic decisions, the management may lay an emphasis on entering or creating new markets, as opposed to increasing market shares in existing markets. It may, on the other hand, suggest a goal of increasing overall market share as a strategic initiative instead of focusing on short term profits [Banfe 1992]. The output of this complex decision process is the airline schedule.

To simplify the analysis, consider a schedule planning paradigm (Figure 1-3) used by many of the major carriers [Talluri and Gopalan 1995]. The decisions in the

paradigm are divided into several modules, each represented by a box. The arrows indicate the sequence of decisions made, and the *feedback* arrow from the last box (crew planning) to the first box (flight scheduling) indicates that this process is iterative and any difficulties in getting the schedule operational have to be sorted out.

Generating a Flight Schedule

In Figure 1-3, *flight scheduling* refers to the process of setting up a schedule of flight legs, where each leg is characterized by an origin city, a destination city, a start time, and an end time. This process must take into account both tactical decisions and strategic goals. An important aspect of flight scheduling is the estimation of travel demand [Ben-Akiva and Lerman 1985]. The estimation of travel demand for each flight leg consists of two steps - *traffic forecasting* and *traffic allocation* [Talluri and Gopalan 1995]. Traffic forecasting refers to the estimation of aggregate demand for an origin-destination market over a period of time like a week, month, or season. Traffic allocation, on the other hand, deals with allocating this demand to specific flight legs. Traffic forecasting and allocation methods are covered in detail in books such as Wells [1994].

Fleet Assignment

After generating a flight schedule, the airline must decide which equipment type, or *fleet type*, must fly each flight. This decision is referred to as the *fleet assignment* problem [Hane et al 1994]. The assignment of fleets to flights can be done using several criterion, such as maximizing revenues, maximizing profits, minimizing costs, or even minimizing the number of aircraft used. The goal is to generate an assignment that optimizes an objective while satisfying a host of requirements, such as the coverage of all flights in the schedule, balance of aircraft at each airport, and availability of aircraft [Abara 1989].

Through Flight Assignment

The airline must next decide which flights to classify as *throughs*, i.e., sequences of flights flown by the same aircraft. This is called *through flight assignment*, and a through flight generates extra revenues over a *connect*, i.e., an itinerary in which the commuters have to change aircraft, because passengers are willing to pay a premium to stay on the same aircraft rather than make a tight connection where the chance of lost luggage is also greater [Talluri and Gopalan 1995].

Aircraft Maintenance Routing

Given fleet assignments and the selected through flights, the airline must determine routes to be flown by each individual aircraft. Safety is an important issue for airlines, and the problem of routing the aircraft so that safety standards are satisfied is called the *aircraft maintenance routing* problem [Talluri and Gopalan 1995]. The fleet assignment, through flight assignment, and aircraft maintenance routing problems have been well studied, and the literature will be covered in later chapters.

Crew Planning

Crew planning refers to creation of schedules for the crew. Crew planning consists of two main steps. In the first step, called *crew pairing optimization*, a minimum cost set of *pairings* that covers all the flights is generated. Pairings are routes that start and end at the same crew base and satisfy a host of contractual and government rules [Barnhart et al 1993]. In the second step, called *crew rostering* or *crew bidding*, particular pairings are assigned to each crew member. In crew rostering, crew members are assigned *rosters* that satisfy a variety of constraints that are specific to each crew member [Gamache et al 1994]. In crew bidding, *bidlines*, i.e., monthly schedules of pairings for a crew, are generated and the crew members show preferences for these schedules by placing bids. Assignment of bidlines or schedules is done based on the value of the bid and crew seniority [Jones 1989]. Extensive work has been done in crew planning, and the literature for this work will also be covered in a later chapter.

1.2.3 Domestic vs. Long-Haul Operations

For U.S. carriers, domestic flight networks, and international, or *long haul* flight networks differ in certain ways [Shenoi 1994, Barnhart 1996]. Specifically,

- Domestic flights are typically shorter and more frequent than long haul flights.
- Domestic flights (for the most part) are scheduled on a daily basis, i.e., on every day of the week. Long haul flights are typically weekly, i.e., if a flight is scheduled on a specific day of the week, it will be scheduled on that same day in *every* week.
- Domestic flights are typically arranged in a *hub-and-spoke* system with many flights arriving at, or departing from, central airports called *hubs*. Long-haul networks are typically *point-to-point* systems.
- In domestic operations, unlike long-haul operations, many flights arrive at, or depart from, central airports, called *hubs*, during periods of high activity called *banks* or *complexes*.
- The U.S. domestic airline industry is deregulated, while the international aviation industry (with a few exceptions) is heavily regulated by bilateral agreements.

These distinguishing properties between domestic and long-haul problems make them amenable to different models and solution procedures.

1.3 Contributions and Outline

The focus of this dissertation is on the development of integrated models that will allow simultaneous, rather than sequential, solution of some of the airline schedule planning decision modules. The benefits of a simultaneous solution process are the following:

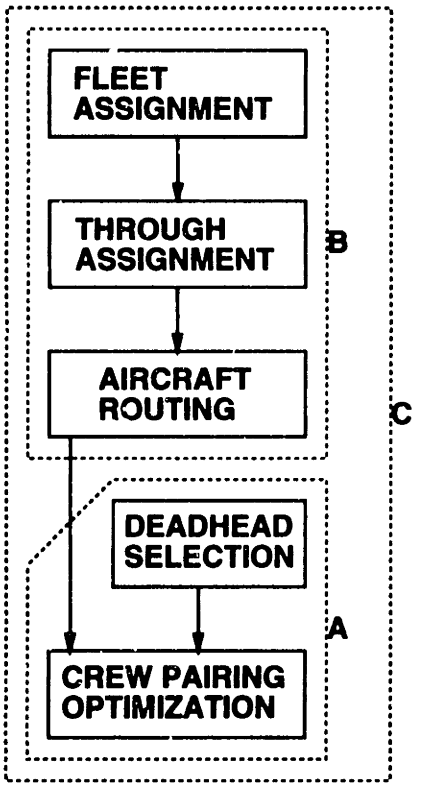

Figure 1-4: Contributions of The Dissertation

1. Simultaneous decision making will produce more economical schedules that may be closer to achieving *global* optimality.

2. In some cases, the sequential solution of decisions will not yield feasible solutions. With simultaneous solutions, incompatibilities between decisions are reduced. This should lead to fewer difficulties in making the schedule operational, thereby allowing a reduction in overall schedule generation time and improved productivity of the schedule developers.

The decision modules that are considered in this work, shown in Figure 1-4, are difficult to solve, either simultaneously or sequentially. Our work makes the following four contributions.

1. *Comprehensive review of literature:* In the following chapters, we provide a comprehensive survey of research that attempts to solve similar problems, or

different instances of these problems.

2. *Design of approximate models and solution strategies for crew pairing problems:* In chapter 3, we study the crew pairing optimization problem and deadhead selection. Deadhead flights play a very crucial role in long-haul crew schedules and reduce crew costs by repositioning crews to increase their utilization. A model is developed that closely approximates the crew pairing problem. This model can be used to select deadheads to reduce crew costs. This same model can also be used to decrease overall solution time. Using our approach, computational experiments with data provided by a long-haul airline show that significantly improved solutions can be achieved quickly. Compared to existing methods, solution costs and run times are reduced by about 20% and 80%, respectively.

3. *Design of integrated models and solution procedures for aircraft scheduling:* For U.S. domestic operations, the fleet assignment, through flight assignment, and maintenance routing decisions typically are made sequentially. However, sequential decisions may not yield legitimate maintenance schedules in international, or *long-haul* operations, due to low schedule frequency and sparse network structure. In fact, using data provided by a long-haul airline, we show that a sequential solution method does not give aircraft schedules that satisfy maintenance requirements. In chapter 4, we present integrated models for the *aircraft scheduling* problem, i.e., the combined fleet assignment, through flight assignment, and aircraft maintenance routing problem. While these models can be used to perform simultaneous fleet assignment, through flight assignment, and aircraft routing, they can also be used to find maintenance routings for a specific fleet given the flights that are to be flown by that fleet.

   In this chapter, we also present specialized column generation and branching strategies that are needed to solve the integrated model. Using our model and the long-haul carrier's data, we obtain near-optimal solutions to the integrated model. Finally, we show how this model can be extended to handle maintenance base planning.

4. *Design of solvable integrated models for fleet assignment and crew scheduling:* On one hand, fleet assignment has a large effect on airline revenues [Subramanian et al 1994]. On the other hand, crew costs are one of the largest components of airline operating costs (Table 1.3) [Simpson and Belobaba 1993[8]], and crew schedules are created separately for each set of flights assigned to a fleet[9]. Given that these two decisions have the largest impact on the schedule development process, simultaneous, rather than sequential, solution could have a tremendous impact on costs.

   The state-of-the-art, however, is still far from solving such a model. In chapter 5, we move in the direction of simultaneous aircraft and crew scheduling by proposing an integrated model building on the concepts developed in chapters 3 and 4. Namely, the approach is to combine approximate models of the crew and aircraft scheduling problems. These approximate models are potentially tractable and good representations of the problem in that our work suggests that the optimal solution value for the approximate integrated model will be very close to the optimal solution value for the original problem. Such models can be used as a first step in generating integrated solutions to the aircraft and crew scheduling problems. We also present conclusions and directions for future research in chapter 5.

[8]The fuel consumption numbers in Tables 1.2 and 1.3 probably differ due to differences in accounting methods used by the two sources.

[9]Crews are actually assigned to a *crew compatible* fleet group, i.e., a set of fleets that differ in the seating arrangements, but that have the same cockpit requirements. Thus, they differ for marketing reasons, but they are identical for crew planning purposes [Clarke et al 1995b].

Type of cost ↓	*% Operating costs (by year)*				
	1970	1975	1980	1985	1992
A Flight Operating Costs (FOC)					
Fuel Costs	12.5	17.9	29.8	21.8	12.9
Pilots Costs	13.0	11.9	11.1	10.7	9.7
Direct Engine costs	7.7	6.6	4.8	4.9	6.1
Indirect (Burden)	6.3	5.1	3.9	3.3	4.0
Depreciation	8.8	6.3	4.0	4.7	5.1
Rentals	2.8	2.3	1.0	1.2	7.8
Insurance (hull)	0.5	0.3	0.1	0.3	0.4
Total FOC	**52.4**	**51.0**	**55.6**	**47.4**	**50.4**
B Ground Operating Costs (GOC)					
Traffic servicing	8.7	9.0	8.0	9.4	10.1
Aircraft servicing	8.0	7.4	5.9	5.9	6.1
Service administration	1.1	1.0	8.0	1.1	1.0
Commisions	2.5	3.4	4.8	7.8	7.6
Reservations & sales	6.6	5.6	5.5	7.7	6.7
Total GOC	**26.9**	**26.4**	**25.1**	**31.9**	**30.7**
C System Operating Costs (SOC)					
Cabin Crew Costs	-	-	5.0	3.9	4.7
Meals	-	-	4.4	3.4	3.5
Advertising	2.5	1.9	1.8	2.3	1.2
General & Administrative	4.5	4.2	3.2	4.2	3.7
Ground Equipment	3.2	3.0	3.5	1.8	2.8
Other	-	-	1.2	1.8	1.2
Total SOC	**20.7**	**22.6**	**19.3**	**20.7**	**18.9**
Total Operating Costs	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**

Table 1.3: US Domestic Major Airline Costs - % Breakdown

Chapter 2

Review of Relevant Methodologies

Most airline schedule planning problems are complicated by constraints on the *sequences* of flights flown by a *resource*, i.e., an aircraft or a crew. These constraints are numerous and difficult to formulate mathematically. This difficulty can be overcome by enumerating all possible sequences, and assigning a decision variable to each sequence to decide whether the sequence is assigned to the resource or not. The advantage of this method is that the sequence restrictive constraints can be eliminated by considering only those sequences that satisfy these constraints. The disadvantage, however, is that the number of variables may be huge. For major airlines, the number of variables measure in the billions [Vance 1993]. This chapter studies methodologies that are commonly used to solve integer problems containing many variables, i.e., *branch-and-bound*, *branch-and-price*, and *column generation*.

2.1 Branch-and-Bound

Branch-and-bound is a *divide-and-conquer* method of solving integer programs (IP's) [Bradley et al 1977]. This method first relaxes the integrality constraints of the IP, and solves the resultant linear program (LP). This LP is referred to as the *root node* LP of the branch-and-bound *enumeration tree*. If the optimal LP solution has no fractional values, the solution satisfies integrality constraints and is also the optimal IP solution. However, if any variable x_j^* in the optimal LP solution is fractional, the

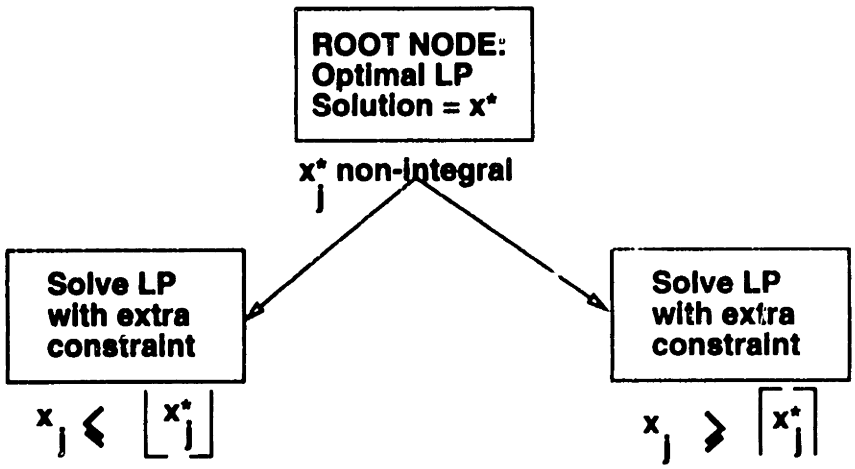

Figure 2-1: Branch-and-Bound Logic

fractional solution is eliminated using a *branching rule*, i.e., a set of constraints that partitions the feasible integer polyhedron into mutually exclusive subdivisions, each of which is the feasible space corresponding to one of the newly created branches.

Since there are many methods of partitioning the feasible region, there are many possible branching strategies. Figure 2-1 shows the root node of the branch-and-bound tree, the optimal LP solution of which has a fractional variable, x_j^*. Without loss of generality, let us assume that the LP is a minimization problem. A branching rule is used that creates two branches for this variable. On the right branch, we force $x_j \geq \lceil x_j^* \rceil$, and on the left branch we force $x_j \leq \lfloor x_j^* \rfloor$. This creates two new nodes in the enumeration tree, and an LP is solved at each of these nodes. The possible results of solving an LP at each node are [Bradley et al 1977]:

1. infeasible LP (node is *fathomed*, i.e., the enumeration tree is pruned since no further exploration is needed at this node);

2. the LP optimal solution is not better than the current best integer solution (node is fathomed);

3. the LP optimal solution is integral and better than the current best integer solution (node is fathomed);

4. the LP optimal solution is fractional, but the solution is better than the current best integer solution (node is kept *active*, i.e., further exploration can be performed from this node).

In the first three outcomes, the node has been fathomed and no further exploration of the node is necessary. The first outcome is infeasibility of the LP which implies that no feasible IP solution exists. Further restrictions on this node will only lead to infeasibilities. The second outcome, in which the LP solution value is greater than the current best integer solution value, indicates that no solution deeper in the tree starting at this node will improve the current solution. This is because the LP optimal solution is a lower bound on the optimal integer solution and the LP solution can only increase with further exploration of this node. The third outcome results in an improved IP solution and futher exploration of this node will not yield a better solution. Again, this follows because the solution quality cannot improve as the problem becomes more restricted. Only the fourth outcome warrants further branching. In this case, further exploration may yield an improved integer solution, and so the node is kept *active*.

2.1.1 A Branch-and-Bound Example

To demonstrate this algorithm, we use an example presented in Bradley et al [1977]. A part of that example is reproduced below and in Figure 2-2. They consider a *maximization* problem of the form:

$$\max z = 5y_1 + 8y_2 \tag{2.1}$$

subject to

$$y_1 + y_2 \leq 6 \tag{2.2}$$

$$5y_1 + 9y_2 \leq 45 \tag{2.3}$$

$$y_1, y_2 \geq 0 \tag{2.4}$$

$$y_1, y_2 \in Z \tag{2.5}$$

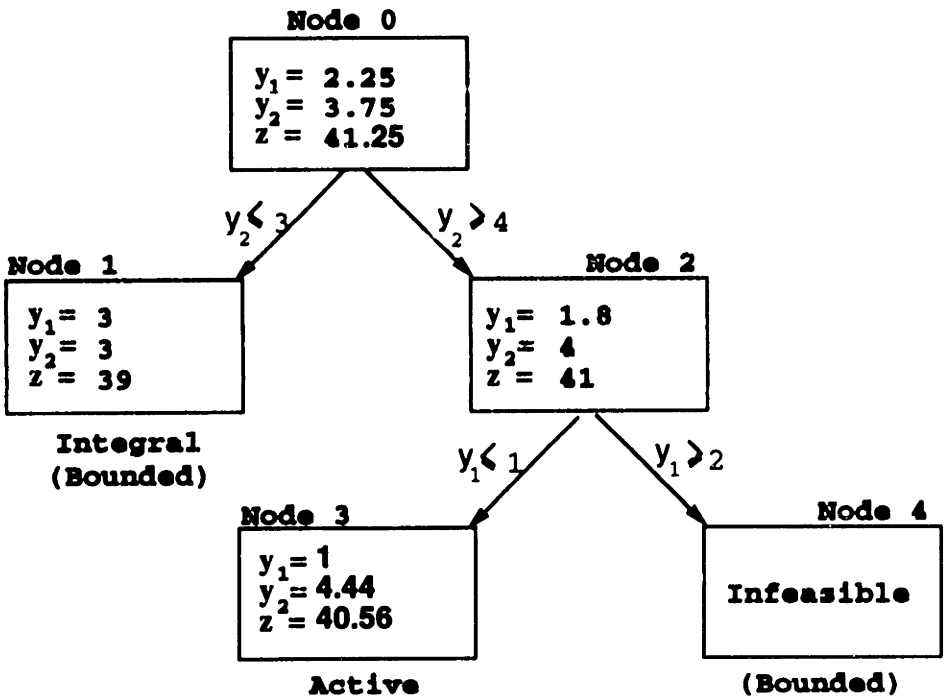

Figure 2-2: (Partial) Branch-and-Bound Example

1. **Root Node Solution:** Integrality requirements are ignored and the optimal LP solution value is $41\frac{1}{4}$ (an upper bound on the optimal IP solution) and y_1 and y_2 are $2\frac{1}{4}$ and $3\frac{3}{4}$ respectively.

2. **Root Node Branching:** Branching on y_2 (chosen arbitrarily) creates two active nodes, nodes 1 and 2, with the added constraints $y_2 \leq 3$ and $y_2 \geq 4$ respectively.

3. **Node 1 Solution:** The optimal LP solution value of node 1 (again chosen arbitrarily) is 39 and y_1 and y_2 have values of 3 each.

4. **Fathom Node 1:** The optimal LP solution of node 1 is both integral and the best IP solution so far, thus providing a lower bound on the global IP solution value. This node can be fathomed.

5. **Node 2 Solution:** The optimal LP solution value of node 2 is 41, and the values of y_1 and y_2 are $1\frac{4}{5}$ and 4 respectively.

6. **Node 2 Branching:** Since y_1 is not integral, branching is performed at node 2, creating node 3 on the $y_1 \leq 1$ branch, and node 4 on the $y_1 \geq 2$ branch.

This procedure continues either until optimality is reached, or some predetermined termination criterion is achieved. Two decisions that are made many times in this algorithm are the *branching decision*, i.e., deciding the variable on which to branch, and *node choice*, i.e, deciding which active node to choose for exploration. In the above example, the decisions were made arbitrarily.

2.2 Column Generation

Although branch-and-bound methods can be used to solve any integer program, computational difficulties arise when the number of variables gets large. In fact, even the solutions of LP's with a large number of variables are difficult and require the use of specialized solution methods such as *column generation* [Bradley et al 1977]. This section discusses the column generation algorithm, while section 2.4 shows how large integer problems can be solved by placing column generation within a branch-and-bound framework to create a solution method called *branch-and-price* [Barnhart et al 1995b].

Column generation methods, based on the decomposition principle of Dantzig and Wolfe [1960, 1961], are based on the fact that the entire set of variables (columns) are unnecessary for solution, and can be generated as needed [Ahuja et al 1993]. This method solves an LP by alternately solving a *restricted master problem* and smaller *subproblems*, which we will refer to as *pricing problems*. Column generation and decomposition are sometimes called generalized linear programming(GLP) [Wolfe in Dantzig 1963, Magnanti et al 1976].

2.2.1 The Column Generation Algorithm

The following LP, called the *master problem* (MP), has a large number of variables $y_1, y_2, \ldots, y_n$, and is shown below [Bradley et al 1977]:

$$z^* = \quad \min z = \quad c_1y_1 \quad +c_2y_2 \quad + \ldots \quad +c_ny_n$$

subject to

$$e_{i1}y_1 \quad +e_{i2}y_2 \quad + \ldots \quad +e_{in}y_n \quad = b_i \quad (i = 1, 2, \ldots, l)$$

$$y_j \geq 0 \quad (j = 1\ 2, \ldots, n)$$

Since this is a large problem, we create a restricted problem with only a subset of the variables, i.e., $y_1, y_2, \ldots, y_m$, and call this the *restricted master problem* (RMP):

$$z^* = \quad \min z = \quad c_1y_1 \quad +c_2y_2 \quad + \ldots \quad +c_my_m$$

subject to

$$e_{i1}y_1 \quad +e_{i2}y_2 \quad + \ldots \quad +e_{im}y_m \quad = b_i \quad (i = 1, 2, \ldots, l)$$

$$y_j \geq 0 \quad (j = 1, 2, \ldots, m)$$

After solving the RMP, we check if the RMP solution is optimal to MP by using the dual prices $\pi_1, \pi_2, \ldots, \pi_l$ (associated with constraints $1, 2, \ldots, l$ respectively) to calculate the reduced cost of variables that have been excluded from the RMP. The reduced cost of variable j is

$$\bar{c}_j = c_j - \sum_{i=1}^{l} \pi_i e_{ij}. \tag{2.6}$$

The RMP solution is optimal to the MP if the reduced cost of each variable is non-negative. This can be checked by solving the following *pricing problem*:

$$w^* = \min_{j=1,\ldots,n} [c_j - \sum_{i=1}^{l} \pi_i e_{ij}]. \tag{2.7}$$

If $w^* \geq 0$, no variables can improve the RMP and the current solution is optimal to the MP. If $w^* < 0$, column k (with $\bar{c}_k < 0$) can potentially improve the RMP. It is added to the RMP and the RMP is solved again. This process is repeated until no negative reduced cost variables are identified, in which case optimality is achieved.

Column generation can be summarized as [Bradley et al 1977]:

- **STEP 0:** Find a subset **R** of variables that is feasible to the MP.
- **STEP 1:** Solve the RMP over **R** and obtain dual prices.
- **STEP 2:** Using these dual prices, solve the pricing problem (equation 2.7) to find a minimum reduced cost column.

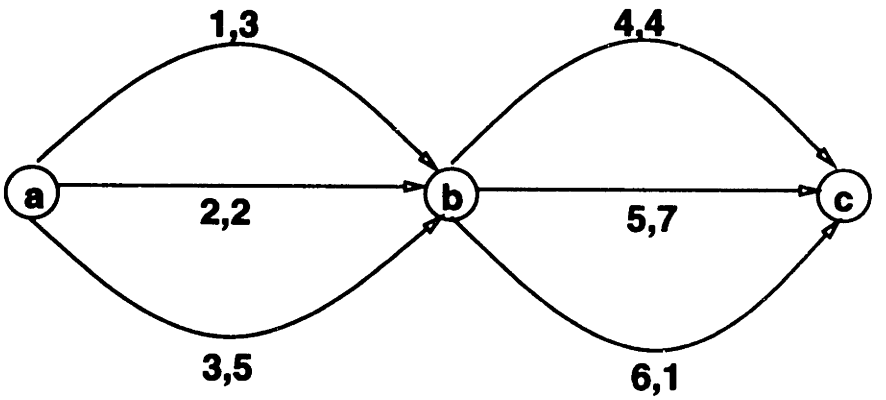

Figure 2-3: Column Generation Example Network

	$7y_1$	$10y_2$	$6y_3$	$9y_4$	$9y_5$	$12y_6$	$4y_7$	$3y_8$	$6y_9$		
1	1	1					1			$\geq$	1
2			1	1				1		$\geq$	1
3					1	1			1	$\geq$	1
4	1		1		1					$\geq$	0
5		1		1		1				$\geq$	0
6							1	1	1	$\geq$	0

Table 2.1: The Master Problem

- **STEP 3:** If the reduced cost of this column is non-negative, then stop - RMP is optimal to MP. Otherwise, add this column to **R** and go back to step 1.

2.2.2 An Example of Column Generation

Consider the network with three nodes and six arcs, shown in Figure 2-3. Each arc has two numbers associated with it, the arc number and the arc cost. There are 9 possible paths between nodes a and c. Suppose the problem is to *find a minimum cost set of paths from node a to node c that cover arcs 1, 2 and 3 at least once.* Although simple inspection will indicate that the optimal solution is a set of three $(a-c)$ paths, $\{1,6\}, \{2,6\}$, and $\{3,6\}$ with a cost of 13, we consider a mathematical formulation to demonstrate the column generation algorithm.

The constraint matrix for the MP (Table 2.1) has 6 rows (one for each arc) and 9 columns (one for each $a-c$ path). Each column has an entry of 1 in a row

	$7y_1$	$10y_2$	$6y_3$	$9y_4$	$9y_5$	$12y_6$		
1	1	1					$\geq$	1
2			1	1			$\geq$	1
3					1	1	$\geq$	1
4	1		1		1		$\geq$	0
5		1		1		1	$\geq$	0
6							$\geq$	0

Table 2.2: The Restricted Master Problem

corresponding to an arc if the path corresponding to that column covers that arc. The cost of each variable is given before each variable name. Assume, for the sake of exposition, that it is not possible to generate the MP completely, and that the initial RMP (Table 2.2) has 6 columns, i.e., paths $\{1,4\}$, $\{1,5\}$, $\{2,4\}$, $\{2,5\}$, $\{3,4\}$, and $\{3,5\}$. The algorithm proceeds as follows:

1. **Step 0** The starting set of columns, R, is chosen (Table 2.2).

2. **Step 1** The RMP is solved to optimality over R giving a solution of 22, variables y_1, y_3, and y_5 are equal to 1 (the rest are zero), and the vector of optimal dual prices is $\{7,6,9,0,0,0\}$.

3. **Step 2** In this example, we find the variable (path in the network) with the minimum reduced cost by *explicitly* scanning each path and calculating its reduced cost. The reduced cost of a path in the network is the sum of the reduced costs of the arcs on that path. The reduced cost of an arc can be calculated by deducting the dual price associated with that arc from the actual cost of the arc. This yields a reduced cost vector of $\{-4,-4,-4,4,7,1\}$ and generates a minimum reduced cost path $\{1,6\}$ whose reduced cost is -3.

4. **Step 3** Since the path $\{1,6\}$ has a negative reduced cost, it is added to R and we go back to step 1.

5. **Step 1** The RMP is solved over R (with 7 columns) to give an optimal solution of 19 (an improvement over 22), y_3, y_5, and y_7 are equal to 1 each (all other

variables are zero), and the optimal dual price vector is $\{4, 6, 9, 0, 0, 0\}$.

6. **Step 2** Reduced costs are calculated as before giving a reduced cost vector of $\{-1, -4, -4, 4, 7, 1\}$. The pricing problem generates path $\{2, 6\}$ which has the minimum reduced cost (-3).

7. **Step 3** Since path $\{2, 6\}$ has a negative reduced cost, it is added to R and we go back to step 1.

8. **Step 1** The RMP is solved over R (8 columns) giving an optimal solution of 16, y_5, y_7, and y_8 are equal to 1 each (all other variables are zero), and the optimal dual price vector is $\{4, 3, 9, 0, 0, 0\}$.

9. **Step 2** The vector of reduced costs is $\{-1, -1, -4, 4, 7, 1\}$. The pricing problem generates a minimum reduced cost path $\{3, 6\}$ whose reduced cost is -3.

10. **Step 3** Since path $\{3, 6\}$ has a negative reduced cost, it is added to R, and we go back to step 1.

11. **Step 1** The RMP is solved over R (9 columns) giving an optimal solution value of 13, y_7, y_8, and y_9 are equal to 1 each (other variables are zero), and the optimal dual price vector is $\{4, 3, 6, 0, 0, 0\}$.

12. **Step 2** The vector of reduced costs $\{-1, -1, -1, 4, 7, 1\}$. The pricing problem generates a minimum reduced cost path $\{1, 6\}$ which has zero reduced cost.

13. **Step 3** Since the minimum reduced cost path $\{1, 6\}$ has a non-negative reduced cost, optimality is achieved.

The integrality of the solutions is coincidental. Column generation is used in a variety of applications such as Airline Crew Pairing Optimization [Barnhart et al 1993], Airline Crew Rostering [Gamache et al 1994], Vehicle Routing [Desrosiers et al 1984], Transit Crew Scheduling [Desrochers and Soumis 1989], Communication Systems Optimization [Ribiero et al 1989, Parker and Ryan 1994], Cutting Stock Problems [Gilmore and Gomory 1961, 1963, Vance 1993], Daily Aircraft Routing and Scheduling [Desaulniers 1994b], etc.

2.3 The Pricing Problem

Column generation approaches use the fact that an LP can be solved without actually including all the variables in the LP. Variables are generated as needed by the pricing problem [Ahuja et al 1993]. In the example above, the pricing problem explicitly *prices-out*, i.e., calculates the reduced costs of, all possible variables. While this approach is feasible for small problems, it becomes intractable for realistic size problems. This section shows how the solution of the pricing problem (Equation 2.7) can be achieved by pricing-out varibles *implicitly*, rather than explicitly, using shortest path procedures.

2.3.1 Underlying Networks

Airline scheduling problems are modeled using networks. Here we describe two such networks, namely the *time-line* network, and the *connection network*.

The Time-Line Network

A time-line network [Hane et al 1994] is one in which each node is associated with both a geographic location as well as a time dimension. If the time-line network is a *flight* network, there is one arc for each flight, and one node for each *event*, i.e., an arrival or departure. In this network, the starting node represents the origin and starting time of the flight, while the ending node represents the destination and *ready time* of the flight. The ready time is the time that the aircraft or crew flying that flight is ready after achieving the minimum connection time at that station.

All nodes at a city are sorted in increasing order of time, and for those nodes with the same time, arrivals are placed before departures. *Ground arcs* are placed between chronologically successive nodes. These arcs, that represent periods of time when the aircraft or crew are on the ground, allow connections between the flight arcs. A special kind of ground arc, called the *wrap around* arc, connects the last event at a station with the first event at that station. These arcs provide continuity of flow of aircraft or crew.

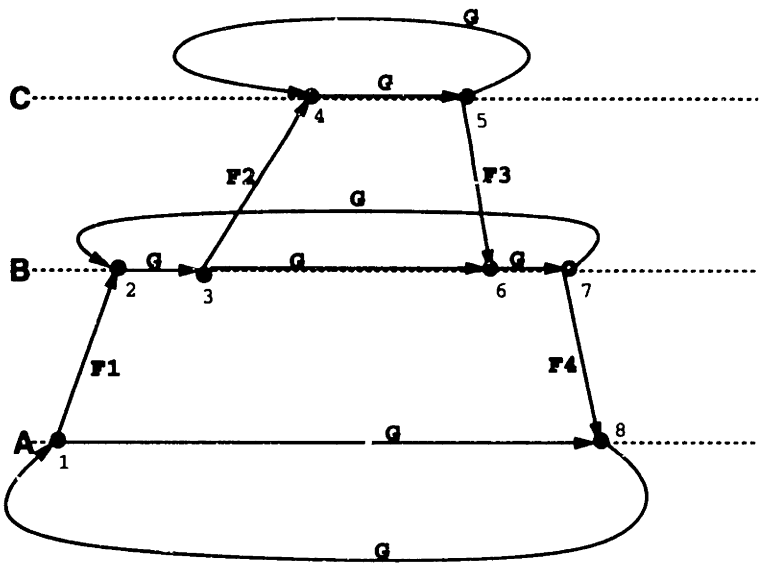

Figure 2-4: A Time-Line Flight Network

Example 1 *Figure 2-4 shows a time-line flight network with four flights,* **F1**, **F2**, **F3**, *and* **F4**, *and three cities* **A**, **B**, *and* **C**. *There are also 8 ground arcs, denoted* **G**, *of which three are wrap-around arcs, one for each city.*

Network Simplification using Node Aggregation

Node *aggregation* or *consolidation* is a method of simplifying time-line networks to make them more compact [Hane et al 1994]. This is based on the observation that a set of arrivals at an airport followed by a set of departures at the same airport can be consolidated into one node without destroying network structure.

Example 2 *Figure 2-5(a) shows an airport with three arrivals and two departures, numbered* **1** *through* **5**. *Nodes* **1**, **2**, *and* **3** *can be aggregated into one event, node* **6**. *Similarly, nodes* **4** *and* **5** *can be aggregated into node* **7**. *Figure 2-5(b) shows that this eliminates three nodes and three ground arcs.*

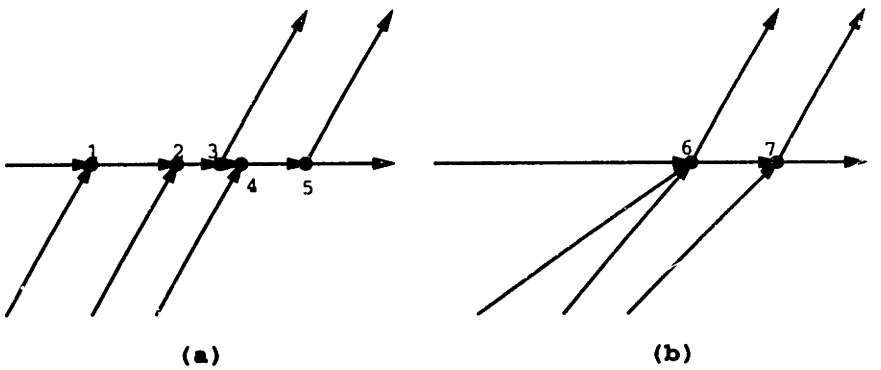

Figure 2-5: Node Aggregation

The Connection Network

Although the time-line network is compact, especially with node consolidation, one of its drawbacks is that a cost or a constraint on a connection between a pair of flights cannot be modeled. For example, in Figure 2-4, it is difficult to enforce a constraint that allows **F1** to connect with **F4** but not with **F2**. This motivates the use of the *connection* network[1], which is similar to the time-line network but has no ground arcs. Instead, there is one arc for each valid connection between pairs of flights.

Example 3 *Figure 2-6 shows a connection network with the same set of flights and cities as in Figure 2-4. In this network, the ground arcs and wrap-around arcs are replaced with 5 connection arcs, all denoted* C.

2.3.2 The Pricing Problem as a Shortest Path Problem

For many airline schedule planning problems, the underlying network characteristics can be exploited to solve the pricing problem efficiently. Since the goal of the pricing problem is to find a variable with the minimum reduced cost, we create an appropriate network in which the variable represents a path, and the cost of the path is the reduced

[1]Such networks are commonly used in the literature under the names *line graph* or *assignment network*, etc. [Lavoie et al 1988, Vance 1993, Talluri and Gopalan 1995].

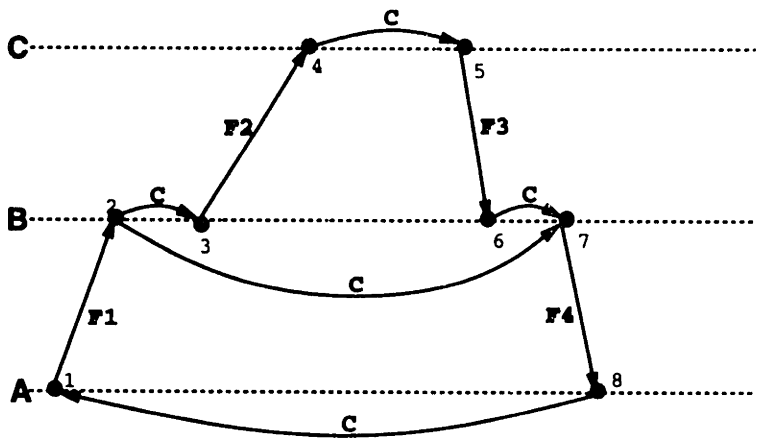

Figure 2-6: A Connection Flight Network

cost of the variable [Ahuja et al 1977]. This is done by modifying the cost of the arcs in this network so that they represent the reduced costs of using those arcs. The variable with minimum reduced cost corresponds to the path that has the least cost, i.e., the shortest path.

The advantage of using shortest path problems for column generation is that a shortest path algorithm can efficiently find the shortest path among *all* possible paths in that network. In other words, a shortest path algorithm considers *implicitly* all the paths in the network in finding the shortest path [Barnhart et al 1993].

Example 4 *Figure 2-7 shows a time-line flight network with three cities,* **A**, **B**, *and* **C**, *and seven flights numbered 1 through 7, the reduced costs of which equal -3, -8, 4, 6, -7, -4, and 9 respectively. Suppose the goal of the pricing problem is to find the path between cities* **A** *and* **C** *with the minimum reduced cost. A shortest path algorithm is run from every node in city* **A** *to every node in city* **C**. *In this case, the minimum reduced cost path between* **A** *and* **C** *is* $2 \rightarrow 4 \rightarrow 6$ *and its reduced cost is -6.*

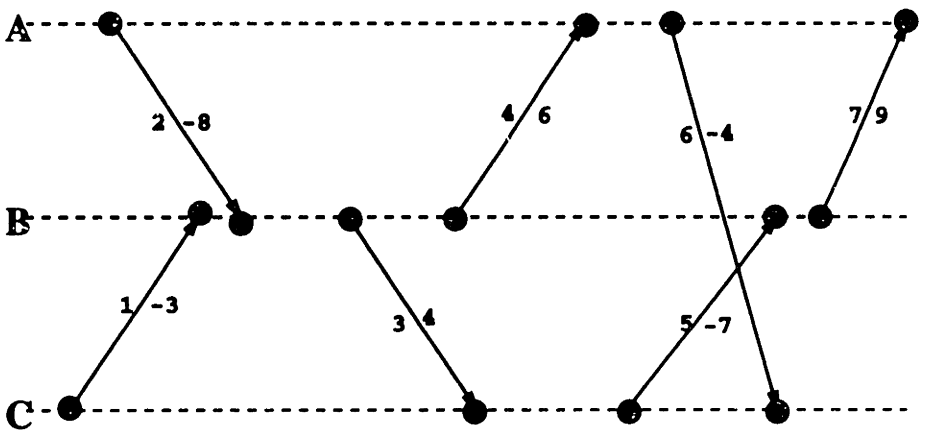

Figure 2-7: A Shortest Path Pricing Problem

2.3.3 Types of Shortest Path Problems

Shortest path problems can fall into several categories depending on the objective and on any requirements placed on the shortest paths [Deo and Pang 1984, Desrochers and Soumis 1988, 1988b, Ahuja et al 1993]:

1. A *simple* shortest path problem refers to one in which there is no requirement placed on the shortest path, and the cost of the shortest path is a simple sum of the costs of the arcs on that path. There is only one cost associated with each arc.

2. A *constrained* shortest path has a restriction imposed on it. Each arc is assumed to consume *resources* in addition to incurring a cost. Examples of resource consumption include time spent or fuel consumed to traverse an arc [Ahuja et al 1993]. The resources used by a path (a sum of the respective resources used by the arcs in that path) are forced to satisfy lower and upper limits on usage. In addition, there may be constraints on subsets of that path. The resources consumed along that path may need to satisfy constraints at each node. For example, *time-window* constraints require a path to arrive at a node during a specified time interval [Solomon 1987].

3. A *multi-criterion* shortest path is one in which the cost of the shortest path is no longer a simple additive sum of the arc costs. Each arc is assigned multiple criterion, i.e., costs, and the cost of a path is a function of the sums of the respective costs on the arcs in the path.

4. A fourth category of shortest paths could be the *multi-criterion constrained* shortest path problem that is a combination of the constrained and multi-criterion shortest path problems.

In simple shortest path problems, the *label* at each node indicates the cost of the current shortest path from the source to that node. Since there are no constraints, and since each arc has only one cost, each label contains exactly one cost. A label can either *dominate* or be dominated by another label, and hence each node need store only one label. Simple shortest path problems can be solved in polynomial time [Ahuja et al 1993].

The difficulty in multi-criterion or constrained shortest path problems is that the cost or feasibility of a path is not known until the path has been completely traversed. It is not trivial to say which label[2] will lead to the shortest feasible path. A label now dominates another label if the costs on one label are all less than or equal to the costs on the other label, and at least one cost on the first label is strictly less than the respective cost on the second. A set of labels, in which no label can dominate any other, is called a set of *efficient* labels [Desrochers and Soumis 1988]. Each label at a node denotes a path to that node. Since all the paths leading to a node can yield efficient labels, and since there are an exponential number of paths in a network, such shortest path problems can take exponential time to solve [Desrochers and Soumis 1988, 1988b].

[2]There is one label at each node for each path leading to that node. This label contains the sum of the costs and resources of the arcs in that path.

2.3.4 Shortest Paths and Airline Scheduling

The shortest path problem is relevant to airline scheduling applications because it can often be used to model the column generation pricing problem. The study of shortest path problems and development of efficient algorithms for their solution is important because such problems are solved repeatedly when solving larger integer problems, especially when column generation methods are employed.

Examples of the use of shortest path methods in solving transportation applications are:

1. Houck et al [1980] use a version of the $n - path$ shortest path problem to solve the travelling salesman problem. An $n - path$ is a path from a vertex to itself that contains exactly n arcs.

2. Kolen et al [1987] use a shortest path problem with time windows to solve a vehicle routing problem.

3. Barnhart et al [1993] and Vance [1993] use shortest path problems with constraints as well as multiple criterion to solve crew scheduling problems.

4. Gamache et al [1994] use a constrained shortest path problem in the solution of an aircrew rostering problem.

5. Desaulniers et al [1994b] use a constrained shortest path problem to solve a daily aircraft routing and scheduling problem.

2.3.5 A Multi-Criterion, Constrained Shortest Path Algorithm

Because shortest path problems and their solutions are so prevalent in solving airline scheduling problems, we describe an algorithm proposed by Desrochers and Soumis [1988, 1988b], to solve the simple, multi-criterion, constrained, and multi-criterion constrained shortest path problems. Consider a network (A, N), where each arc $a \in A$ is associated with I costs $(c_a^i,\ i = 1, \ldots, I)$ and J resources $(t_a^j,\ j = 1, \ldots, J)$.

Consider a path P whose cost C_P is a function of the sum of the respective arc costs, i.e.,

$$C_P = F\{C_P^1, C_P^2, \ldots, C_P^I\} \tag{2.8}$$

where

$$C_P^i = \sum_{p \in P} c_p^i \;\; \forall i = 1, \ldots, I. \tag{2.9}$$

Similarly, the resource consumed by a path is given by

$$T_P^j = \sum_{p \in P} t_p^j \;\; \forall j = 1, \ldots, J. \tag{2.10}$$

In this problem, we enforce lower and upper bounds on the resources consumed by a path at each node on that path. If T_P^{jk} denoted the amount of jth resource consumed by path P at node k, we enforce the rule

$$L_{jk} \leq T_P^{jk} \leq U_{jk} \;\; \forall j = 1, \ldots, J \tag{2.11}$$

where L_{jk} and U_{jk} are the upper and lower bounds on the consumption of the jth resource at node k. Before describing the algorithm, we describe its important features.

Label Definiton

Each label l in the algorithm has $I + J$ elements. The first I represent the sum of the respective costs of the arcs in the path represented by the label l, and the last J represent the amount of each resource consumed along that path.

Algorithm Initialization

Each node can be associated with a set of efficient labels. Initially, the list will be empty at all nodes. Insert label l_s into the (empty) list at source node s.

$$l_s^i = 0 \;\; \forall i = 1, \ldots, I + J \tag{2.12}$$

Insert one label l_h into the lists at all other nodes.

$$l_h^i = \infty \;\; \forall i = 1, \ldots, I + J \tag{2.13}$$

Extend Label

Consider a node $a \in A$ and an arc $n \in N$ that emanates from node a and goes to node $b \in A$. To extend label l_a from node a to node b, we create label l_b using

$$l_b^i = l_a^i + c_n^i \;\; \forall i = 1, \ldots, I \tag{2.14}$$

$$l_b^{(j+I)} = l_a^{(j+I)} + t_n^j \;\; \forall j = 1, \ldots, J \tag{2.15}$$

Equations (2.14) and (2.15) extend the costs and the resource consumption elements respectively.

Feasibility Check

A label l_b at node b is feasible if it satisfies the resource constraint requirements.

$$L_{jb} \leq l_b^{(j+I)} \leq U_{jb} \;\; \forall j = 1, \ldots, J \tag{2.16}$$

Dominance Check

Label l_b dominates l_c if

$$l_b^i \leq l_c^i \;\; \forall i = 1, \ldots, I \tag{2.17}$$

with at least one i^* such that

$$l_b^{i^*} < l_c^{i^*} \;\; 1 \leq i^* \leq I \tag{2.18}$$

The Algorithm

To complete the algorithm, we need to maintain a set S of *active* labels, i.e., labels that have not yet been *extended* and that can affect the solution. The algorithm is:

1. **Initialize:** The label list at each node is empty. Initialize the list at each node using Equations (2.12) and (2.13).

2. **Pick Label:** From S, the list of active labels, we choose an active label l_a (at node a) for exploration from the S.

3. **Extend Label:** Scan all arcs emanating from node a and extend the label l_a to the node at the end of each of these arcs using Equations (2.14) and (2.15).

4. **Check Feasibility:** Consider a label l_b (node b) that has been extended from label l_a (node a). Check the feasibility of l_b using Equation (2.16).

5. **Check Dominance:** If label l_b is feasible, then use Equations (2.17) and (2.18) to check if l_b can dominate any other labels at node b or if any label at b can dominate label l_b. Delete all dominated labels. If l_b is not dominated, add it to S, the list of active labels.

6. **Check S:** If the active list is empty, STOP. Else go to step 2.

When the algorithm ends, each node (with the exception of the source node s) has a set of efficient feasible labels. To calculate the shortest path cost from node s to any node a, we scan the list of labels at a, and calculate the cost of each using Equation (2.8). The minimum cost gives the cost of the shortest feasible path from s to a. By keeping track of the predecessor node, predecessor arc, and predecessor label, the actual shortest path can be traced. We demonstrate this with a detailed example.

2.3.6 Bi-Criterion Constrained Shortest Path Example

The network given in Figure 2-8 has 5 nodes, denoted **A** through **E**, and 8 arcs. Each arc is associated with two costs and one resource consumption value. For example, the arc from A to B has costs 1 and 2, and uses 3 units of a resource. Each arc is associated with a resource *window*. For example, node D has a window of [6,8] implying that any path has to consume between 6 and 8 units of the resource by the

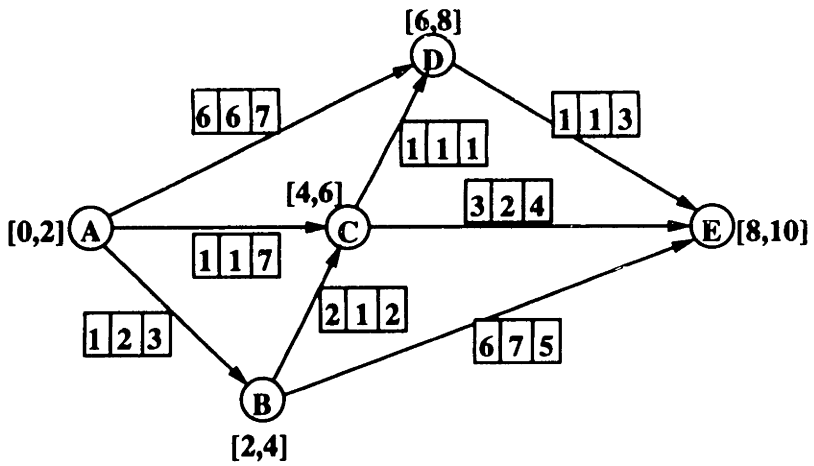

Figure 2-8: A Shortest Path Example

time it reaches node D. The cost of a path is the maximum of the sums of the costs of the arcs in that path. In this example, *we try to find a minimum cost path, from node* **A** *to node* **E**, *that satisfies the resource window constraints at each node.*

To solve this problem, we use the multi-label algorithm described in section 2.3.5. Each label contains 3 elements, two cost elements and one resource consumption element. The algorithm progresses as follows:

1. **Initialize:** Source node A is assigned label (0,0,0) which is inserted into S, the list of active labels. All other nodes are assigned labels (∞, ∞, ∞).

2. **Pick Label:** The only label in S is (0,0,0) at node A.

3. **Extend Label:** Label (0,0,0) at node A is extended to (1,2,3) at node B, (1,1,7) at node C, and (6,6,7) at node D.

4. **Feasibility Check:** Labels (1,2,3) and (6,6,7) are feasible because the resources used by these labels (3 and 7 respectively) fall within the resource windows at these respective windows. Label (1,1,7) is not feasible at node C because it uses 7 units of the resource and the resource window constraint is [4,6].

5. **Dominance Check:** (1,2,3) dominates (∞, ∞, ∞) at node B and (6,6,7) dominates (∞, ∞, ∞) at node D. Both labels are added to set S.

6. **Check S:** S has two labels (1,2,3) and (6,6,7). Go to step 2.

7. **Pick Label:** Label (1,2,3) is chosen (arbitrarily).

8. **Extend Label:** Label (1,2,3) is extended to create (3,3,5) at node C and (7,9,8) at node E.

9. **Feasibility Check:** Labels (3,3,5) and (7,9,8) are feasible at nodes C and E respectively.

10. **Dominance Check:** Label (3,3,5) dominates (∞, ∞, ∞) at C and label (7,9,8) dominates (∞, ∞, ∞) at node E. Label (3,3,5) is added to S, but (7,9,8) is not because that is the final node (for which the shortest path from A is needed) and adding it to S will not help in any way.

11. **Check S:** S contains (3,3,5) and (6,6,7). Go to Step 2.

12. **Pick Label:** (3,3,5) is chosen (again arbitrarily)

13. **Extend Label:** (3,3,5) is extended to give (4,4,6) at node D and (6,5,9) at node E.

14. **Feasibility Check:** Both labels (4,4,6) and (6,5,9) are feasible at their respective nodes.

15. **Dominance Check:** (4,4,6) dominates (6,6,7) at node D, and (6,5,9) dominates (7,9,8) at node E. (4,4,6) is added to S.

16. **Check S:** S contains only (4,4,6). Go to Step 2.

17. **Pick Label:** Choose (4,4,6)

18. **Extend Label:** (4,4,6) is extended to give (5,5,9) at node E.

19. **Feasibility Check:** (5,5,9) is feasible at node E.

20. **Dominance Check:** (5,5,9) dominates (6,5,9) at node E. Since this is node E, the labels are not added to S.

21. **Check S:** S is empty. STOP.

The only label at node E is (5,5,9) whose cost is 5 units. This is the cost of the shortest feasible path from A to E. By retracing the path using the labels, the shortest feasible path is $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E$.

In particular, we would like to highlight a couple of points. In the first feasibility check (calculation 4), label (1,1,7) is infeasible because it violates the resource window constraint. This prevents the generation of two paths $A \rightarrow C \rightarrow E$ and $A \rightarrow C \rightarrow D \rightarrow E$. Thus, by early pruning of infeasible labels, efficiency is increased.

Secondly, in the feasibility check (calculation 9) above, (7,9,8) is feasible at node E. If the goal of the algorithm was just to obtain a feasible path, we could stop the process at this point. This is especially useful during the early stages of the column generation algorithm where the emphasis may be more on getting feasible paths rather than shortest paths. In such cases, we believe it makes sense to run fast approximate shortest path algorithms (e.g., the simple shortest path algorithm) in the early iterations leaving the exact (more time consuming) runs for later iterations [Barnhart 1996]. If the simple shortest path algorithm generates infeasible paths, these paths are simply ignored and the exact methods may be invoked.

The progress of the shortest path algorithm on the network is shown in Figure 2-9. **DOM** implies "dominated" and **INF** implies infeasible.

2.4 Branch-and-Price

Branch-and-price is a method used to solve mixed integer programs (MIP's) when the number of variables is too large to consider explicitly [Barnhart et al 1995b]. As noted by Appelgren [1969], this is a non-trivial extension of branch-and-bound since the LP at each node of the enumeration tree has to be solved using implicit enumeration methods such as column generation and the branching rules should not

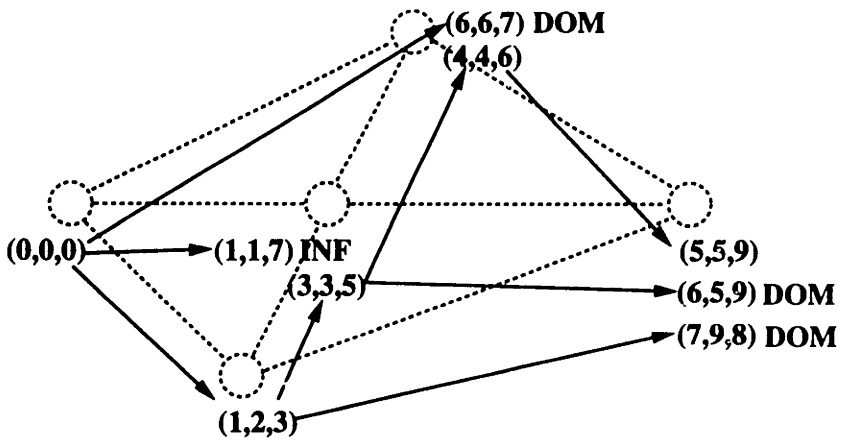

Figure 2-9: Progression of Shortest Path Algorithm

x_1	x_2	x_3	
1	1		$= 1$
	1	1	$= 1$
1		1	$= 1$

Table 2.3: Need for Branch-and-Price

make the column generation pricing problem difficult to solve.

One heuristic approach is to solve only the LP at the root node using column generation and solve the remaining LP's using only the columns generated at the root node. The problem with such an approach is that the optimal root node LP matrix (restricted master problem) contains only a *subset* of all possible variables, and may not contain the variables in the optimal solution to the MIP. In fact, the optimal restricted master problem matrix may not even contain a feasible MIP solution.

Example 5 *Consider a MIP for which the matrix in Table 2.3 is the optimal restricted master problem for the LP relaxation of the MIP. Although the restricted master problem has a feasible LP solution ($x_1 = x_2 = x_3 = 0.5$), it will have no feasible MIP solution until more variables (columns) are added.*

Barnhart et al [1995b] provide a framework and brief survey of branch-and-price

methods. This method has been used to solve problems such as:

1. routing with time windows [Desrosiers et al 1986],
2. crew scheduling [Barnhart et al 1993, Vance 1993],
3. cutting stock problems [Vance 1993],
4. generalized assignment problems [Savelsbergh 1993],
5. airline crew rostering [Gamache et al 1994], and
6. daily aircraft routing and scheduling [Desaulniers 1994b].

Branch-and-price consists of two parts - solving the LP at each node, and branching.

2.4.1 Solving the Branch-and-Price LP

The LP at each node in the branch-and-price enumeration tree is solved using column generation (section 2.2). This requires the repeated solution of the restricted master problem and the pricing problem. As in branch-and-bound (section 2.1), there are four possible outcomes of the solution of the LP at each node. The first three - in which the LP solution is either infeasible, or fractional but not better than the current best IP solution, or integral and better than the current best IP solution - result in the node being fathomed with no further exploration required at these nodes. The last outcome is when the optimal LP solution is non-integral, but is better than the current best IP solution. The best IP solution may therefore be improved by further exploration of the node. This node is kept active for further exploration.

If the list of active nodes is empty, or some predetermined termination criterion is satisfied, then the algorithm is stopped. Otherwise, a node is selected for branching from the list of active nodes.

2.4.2 Branching

After branching has been done at a node, the LP at each of the new branches is solved using column generation. The biggest challenge of branch-and-price is to maintain

compatibility between the rules used for branching and the pricing problem [Barnhart et al 1995b]. This means that a branching rule should be enforceable without changing the structure of the pricing problem.

To illustrate, consider a problem in which a set of arcs must be covered by a set of paths, and each arc must be covered by exactly one path. If the solution to the LP at a node in the branch-and-price tree is integral, then we are done. Otherwise, a (path) variable with a fractional value is selected and a branching rule is invoked creating two branches - one sets the path variable to unity, and the other sets the path variable to zero. These branching rules are enforced as follows:

1. *Set path variable p to unity:* First, set the upper and lower bounds of variable p to unity. Since each arc can be assigned to only one path, the arcs in the selected path p cannot be used by any other path. So, set the upper and lower bounds of any other variable in the constraint matrix containing an arc in path p to zero. Next, eliminate each arc in p from the network by setting their costs to a very large value. In this way, when we solve the pricing problem, the generated paths with negative reduced cost will not contain any arc used by path p.

2. *Set path p to zero:* While it is easy to effectively eliminate the variable p by setting its upper and lower bounds to zero, it is not easy to ensure that the solution to the pricing problem will not regenerate path p since there is no straightforward way to change the network to ensure this. For example, it is not feasible to change the costs of the arcs used by p to large values because this would also eliminate other paths that use the same arcs.

In light of the difficulty to set path p to zero, we alter the branching rules so that (pairs of) arcs, and not paths, are forced in or out of the solution [Barnhart et al 1995b]. This translates into a branching rule that either forces arc i to be followed by arc j in a path, or disallows arc i to be followed by arc j. Both branches are easy to enforce.

1. In one branch (in which arc i is followed by j), the constraint matrix is modified by setting the upper and lower bounds to zero for any path in which arc i is not

followed by arc j or in which arc j is not preceeded by arc i. In this network, all connections from arc i that do not lead to arc j and all connections to arc j that do not start at arc i are eliminated by setting their costs to a very high value. This ensures that the pricing problem solution will always be a path in which i is followed by j, if either i or j is contained in the path.

2. In the other branch, (in which arc j must not follow arc i), the upper and lower bounds are set to zero for all paths in which arc i is followed by arc j. The network is modified by placing a high cost on the connection between arcs i and j. Then, the pricing problem solution will not generate a path in which j follows i.

To summarize, branch-and-price techniques rely on branching strategies that

1. exclude the current fractional solution,
2. create a valid partition of the solution space,
3. guarantee an optimal integer solution with a finite number of branches, and
4. yield tractable pricing problems in which branching decisions can be enforced [Barnhart et al 1995b].

Chapter 3

Crew Pairing and Deadheading

The long-haul crew pairing problem requires the coverage of a set of long-haul flights by a minimum cost set of crew pairings [Barnhart et al 1993]. A crew pairing is a sequence of flights flown by one crew, starting and ending at the same crew domicile, and satisfying a variety of Federal Aviation Administration rules and collective bargaining agreements between the crews and the airline. This chapter presents an alternate model and solution approach for this problem.

3.1 Introduction

A pairing is a set of daily work activities, called duty periods or duties, separated by rest periods [Barnhart et al 1993]. In the United States, the structure of *legal* duty periods and pairings are defined by the Federal Aviation Administration (FAA) and by collective bargaining agreements between crews and airline management. For duty periods, restrictions often include limits on the maximum and minimum time that a crew is idle between consecutive flight legs; a limit on the maximum elapsed time from start to end; and a limit on the total number of hours of flying time, called *block* time. The cost of a duty period is typically expressed as the maximum of three quantities [Vance 1993]. The first quantity is the *flying time cost* determined by multiplying the total flying time in the duty by a cost per unit time. The second cost is the *elapsed time cost* expressed as the total elapsed time of the duty multiplied

by a different (lower) cost per unit time. The third cost is the *minimum guarantee cost* which is equal to some predetermined value. Duty costs, then, are the *maximum* of flying time, elapsed time, and minimum guarantee costs.

For pairings, FAA and collective bargaining restrictions may include limits on the maximum number of duty periods in the pairing; a limit on the minimum number of hours of rest between duties-- this limit may vary depending on the amount of flying time and previous rest received; and a limit on the maximum elapsed time, called *time-away-from-base* [Barnhart et al 1993]. Every pairing is constructed so that a single crew can *legally* perform all the duties and rest periods it contains. The cost of a pairing is also often expressed as the maximum of three quantities. The first quantity is the total cost of the duties contained in the pairing, the second is the cost associated with the pairing's total time-away-from-base, and the third equals the minimum guarantee cost for the pairing [Vance 1993].

Given a set of scheduled flights flown by a particular fleet or aircraft type, the objective of the long-haul crew pairing problem is to find the subset of pairings that covers each flight (i.e., each flight is assigned to a crew) and minimizes total crew costs [Barnhart et al 1993]. Since every solution must assign one crew to each flight, a lower bound on total crew costs is the total flying time costs for the scheduled flights. Costs in excess of this lower bound are referred to as *penalty costs*. The term *pay-and-credit* refers to penalty costs expressed as a percentage of total flying time costs for the scheduled flights [Barnhart et al 1995]. For example, a solution with pay-and-credit of 20% is a solution whose cost is 1.2 times the schedule's total flying time costs.

Our study of this problem is motivated by the large potential for crew cost reductions in current long-haul operations. For example, Barnhart et al [1995] solves problems where the pay-and-credit is as high as 81%, and a 1% improvement in pay-and-credit amounts to savings of about $250,000 annually.

3.1.1 Background

The crew pairing problem (for domestic U.S., medium-haul, and long-haul operations) has been extensively studied [e.g., Thiriez 1969, Arabeyre et al. 1969, Rubin 1973, Gerbracht 1978, Baker et al. 1979, Marsten et al. 1979, Shepardson and Marsten 1980, Baker and Fisher 1981, Baker 1981b, Marsten and Shepardson 1981, Odier, Lascaux and Hie 1983, Minoux 1984, Ball and Roberts 1985, Etschmaier and Mathaisel 1985, Rannou 1986, Crainic and Rousseau 1987, Lavoie, Minoux and Odier 1988, Desrochers and Soumis 1989, Gershkoff 1989, Barutt and Hull 1990, Anbil et al. 1991, 1991b, 1992, 1993, Barnhart et al. 1993, Desrosiers et al. 1991, 1992, Beasley and Cao 1993, Graves et al. 1993, Hoffman and Padberg 1993, Vance 1993, Shenoi 1994, Desaulniers 1994,1995, Wedelin 1995, Atamturk et al. 1995]. While earlier methods are based on heuristics, some recent methods are exact.

In most solution methods, the input is fixed *a priori*. Figure 3-1 shows that most solution methods for the crew pairing problem take flights, designated as *operational* or *deadhead*, as fixed inputs. Operational flights refer to the scheduled flights that must be assigned to a crew, while deadhead flights, or deadheads, are optional and need not be flown [Barnhart et al 1995]. A crew is assigned to a deadhead flight as passengers, either to reposition them so they can be better utilized, or to return them home. Unlike domestic U.S. operations in which deadheading may be limited, deadheading is an essential component in long-haul operations where relatively few flights may be scheduled in and out of a particular location [Barnhart et al 1995].

Barnhart, et al [1995] have applied their deadhead selection procedure to the long-haul crew pairing problem of a U.S. carrier and achieved reductions of over 27% in pay-and-credit by choosing better deadheads. Their approach, depicted in Figure 3-2, iterates between solving the crew pairing problem LP relaxation with some subset of deadhead flights and selecting additional deadhead flights. At each iteration, the selected deadheads are added to the original set of flights input to the crew pairing model. The enlarged LP is then re-solved. The process repeats and is terminated when the computational time reaches some pre-determined maximum

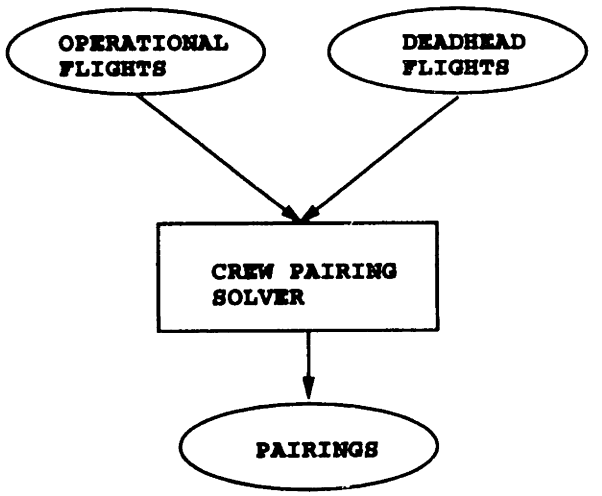

Figure 3-1: Earlier Work on Crew Pairing

or the improvement in solution quality in successive iterations is sufficiently small. Then, the crew pairing IP is solved, allowing any of the selected deadhead flights to be used.

Typically, since the crew pairing LP solver is computationally expensive, the process is terminated by the maximum computing time requirement. In this chapter, we show how this time can be dramatically reduced, enabling many more deadheads to be selected, significantly improving the quality of the crew pairing solutions.

3.1.2 Contributions

In this chapter, we develop a new *inexact* model and solution approach for the crew pairing problem. Our computational experiments have the following results.

1. Our new formulation can be used with a deadhead selector to choose deadheads efficiently. With the new deadheads, the quality of the crew pairing solution can be improved significantly.

2. Our new model and solution approach can be used to speed up solution of exact methods for the crew pairing problem by

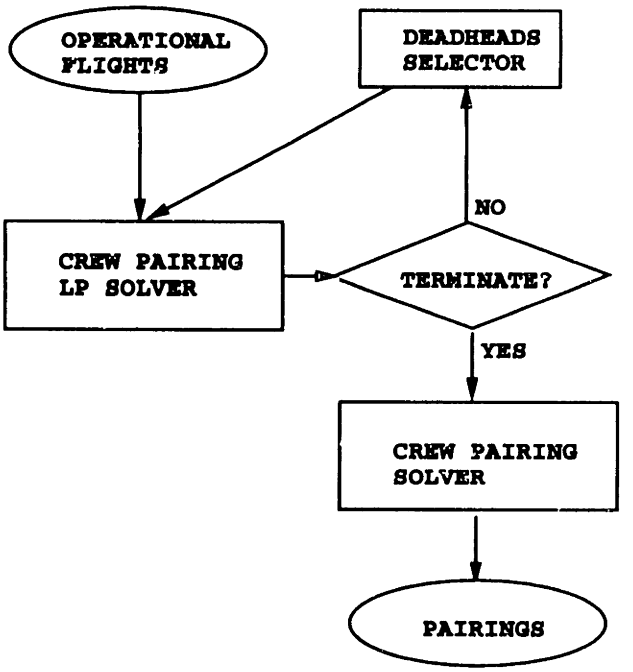

Figure 3-2: Work of Barnhart, Hatay, and Johnson[1995]

- allowing a good initial solution to be generated quickly, and
- quickly providing good lower bounds to the crew pairing problem. These bounds can be used to speed up column generation methods such as branch-and-price (see section 2.4) either by conventional fathoming of the tree, or by terminating the column generation solution of the LP when the gap between our bound and the LP solution is small.

3. Our model's ease of solution allows it to be used for fast scenario analyses.

Our method improves upon the results achieved by Barnhart, et al.[1995] by replacing their crew pairing LP model and solver by a new *approximate* crew pairing model and solver (Figure 3-3). Our new model can be solved quickly and with time limitations, more deadhead selection iterations can be run. The outcome is that better deadheads are identified and the modified input to the crew pairing solver allows improved crew pairing solutions to be generated.

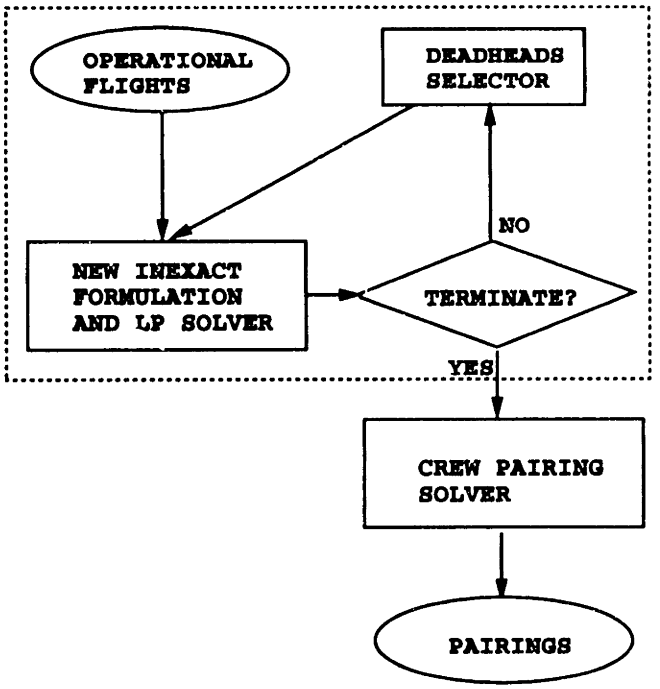

Figure 3-3: Our Decision Flow

An advantage of our method is that it can either replace existing crew pairing solvers or it can be used as a deadhead selection preprocessor (the dotted box in Figure 3-3.)

3.2 Exact Formulation Methods

The crew pairing problem requires the coverage of a set of operational flights by a minimum cost set of pairings. This naturally lends itself to a set partitioning formulation [Garfinkel and Nemhauser 1969]. In the formulation, each variable represents a *valid* pairing. This eliminates the need for explicit constraints capturing maximum time-away-from-base, minimum rest, maximum duty time, and so on. The set partitioning constraints ensure that each flight is covered exactly once.

We consider a simplified model of the crew pairing problem, one in which the cost structure is approximated. Namely, rather than setting the cost of a pairing as the maximum of duty, time-away-from-base, and minimum guarantee costs, we assume

that the cost of a pairing is its time-away-from-base cost. In so doing, we ensure that the optimal solution value we generate is always a lower bound on the true optimal solution value.

We also simplify the problem, as do many crew pairing models, by ignoring certain specialized constraints, like "8 in 24" rules[1], crewbase constraints, etc. We place a restriction, however, on the maximum elapsed time of a pairing. Our resulting formulation for the long-haul crew pairing problem, denoted CPP, is [Barnhart et al 1995]:

$$(CPP) \quad \min \sum_{j=1}^{n} c_j w_j \tag{3.1}$$

subject to

$$\sum_{j=1}^{n} a_{ij} w_j = 1 \qquad i = 1, \ldots, m \tag{3.2}$$

$$w_j \in \{0, 1\} \qquad j = 1, \ldots, n \tag{3.3}$$

where

m = number of flight segments,

n = number of crew pairings,

a_{ij} = 1 if operational flight i is covered by the pairing j and 0 otherwise,

c_j = time-away-from-base cost of crew pairing j,

w_j = 1 if a crew is assigned to pairing j and 0 otherwise.

Equation (3.3) shows that we have one binary variable for each valid pairing. The objective function (3.1) tries to minimize the cost of the pairings that are used, and constraints (3.2) ensure that each operational flight is covered exactly once.

This problem has a large number of variables all of which must be binary. To solve this problem exactly, branch-and-price must be used (section 2.4). Sometimes,

[1]CPP can, in principle, satisfy the "8 in 24" rule by eliminating those pairings, generated by the pricing problem (section 2.3), that violate this rule [Johnson 1996].

heuristic branch-and-price techniques are used. For example, in the experiments of Barnhart, et al [1995], they performed column generation only at the root node of the tree and were able to generate good crew pairing solutions and dramatically reduce computation time.

Observe that flights flown as deadheads are not included in the CPP formulation. This implies that for each operational flight contained in a pairing, a priori decisions are made as to whether the flight will be used in that pairing as a deadhead. This results in a significant increase in the number of decision variables. To illustrate, a pairing j containing operational flights i and l may be a column with $a_{ij} = a_{lj} = 1$, or a column where i is a deadhead flight with $a_{ij} = 0$ and $a_{lj} = 1$, or finally, a column where l is a deadhead flight with $a_{ij} = 1$ and $a_{lj} = 0$. In some cases, this explosion in the number of variables may be averted. For example, if the total cost of deadheading on an operational flight i is some fixed quantity δ_i, then letting z_i represent the number of pairings containing flight i as a deadhead, the CPP formulation[2] can be rewritten as [Barnhart 1996]:

$$\min \sum_{j=1}^{n} c_j w_j + \sum_{i=1}^{m} \delta_i z_i \tag{3.4}$$

subject to

$$\sum_{j=1}^{n} a_{ij} w_j - z_i = 1 \qquad i = 1, \ldots, m \tag{3.5}$$

$$w_j \in \{0, 1\} \qquad j = 1, \ldots, n \tag{3.6}$$

$$z_i \geq 0 \qquad i = 1, \ldots, m \tag{3.7}$$

The w variables will guarantee the z variables to be integer. If the cost of a crew flying flight i equals the cost of a crew deadheading on i , (this is sometimes the case when revenue generating passengers are not displaced by deadheading crews, such as in cargo operations), the CPP problem can be modeled as a set covering problem

[2]This formulation cannot handle certain rules such as allowing duties to exceed limits if the last leg is a deadhead. The formulation may therefore, ignore some feasible duties [Barnhart 1996, Johnson 1996].

[Baker 1981b]:

$$\min \sum_{j=1}^{n} c_j w_j \tag{3.8}$$

subject to

$$\sum_{j=1}^{n} a_{ij} w_j \geq 1 \qquad i = 1, \ldots, m \tag{3.9}$$

$$w_j \in \{0, 1\} \qquad j = 1, \ldots, n \tag{3.10}$$

The set covering formulation [Baker 1981b] has the advantage that it is often easier to solve than the set partitioning formulation.

3.3 A New Model Based on Duties

We develop an approximate model for the long-haul crew pairing problem by relaxing the maximum time-away-from-base constraints in CPP. The purpose is to be able to quickly generate CPP solutions. We achieve this in our model by replacing pairing variables with duty variables. Underlying our formulation is a time-line network of duties [Barnhart, et al 1993], in which each network node is associated with a geographic location as well as time, and arcs represent either duties or periods of stay on the ground at airports. The network is acyclic and each duty arc contains rest time so that minimum rest is achieved at the end of each duty. Our duty period based formulation, denoted DPP, is:

$$(DPP) \qquad \min \sum_{k=1}^{K} \sum_{j=1}^{n} c_j^k x_j^k + \sum_{k=1}^{K} \sum_{j=1}^{p} d_j^k y_j^k \tag{3.11}$$

subject to

$$\sum_{k=1}^{K} \sum_{j=1}^{n} a_{ij} x_j^k \geq 1 \; i = 1, \ldots, m \tag{3.12}$$

$$\sum_{j=1}^{n} b_{lj} x_j^k + \sum_{j=1}^{p} b_{lj} y_j^k = 0 \; l = 1, \ldots, q; k = 1, \ldots, K \tag{3.13}$$

$$x_j^k \in \{0,1\}\ j = 1,\ldots,n; \forall k \in K \quad (3.14)$$

$$y_j^k \in \mathcal{Z}^+\ j = 1,\ldots,p; \forall k \in K \quad (3.15)$$

where:

n = number of duties

m = number of operational flights

p = number of ground arcs

q = number of network nodes

K = number of crew types (i.e., number of crew bases)

x_j^k = 1 if duty j is flown by crew type k; 0 otherwise

y_j^k = number of type k crews using ground arc j

c_j^k = elapsed time cost of duty j flown by crew type k

d_j^k = elapsed time cost of ground arc j when used by crew type k

a_{ij} = 1 if duty j covers operational flight i; 0 otherwise

b_{lj} = 1 if duty j /ground arc j enters node l;
-1 if it leaves node l;
and 0 otherwise

DPP is a multi-commodity min-cost flow formulation where the crew from each crew base are a commodity. For each crew base, we have a variable defined for each duty arc and each ground arc. The cost for each variable is its elapsed time. The objective function (3.11) minimizes the total elapsed time, or equivalently, total time-away-from-base costs. Bundle constraints (3.12) ensure that each operational flight is covered at least once. Flow balance constraints (3.13) ensure conservation of flow for each commodity, i.e., crew type. Constraints (3.14) and (3.15) ensure that the flow on the duty arcs and ground arcs remain binary and integral respectively.

DPP is equivalent to the crew pairing problem formulation CPP except that maximum time-away-from-base constraints may be violated in the DPP solution[3]. DPP, with its significantly reduced size, is easier to solve than CPP. While CPP has only m rows, it has a huge (often measuring in the billions [Vance 1993]) number of columns.

[3]Constraints such as "8 in 24", that can be modeled in CPP are not easy to model using DPP.

DPP, on the other hand has $m + cK$ rows, but only $(n + p)K$ columns. Two observations allow us to simplify the solution of DPP:

1. We can relax the integrality on the ground variables without affecting the solution. The integrality of the duties will result in the integrality of the ground arcs.

2. The variables corresponding to duties containing only deadheads, will not appear in the flight cover constraints. The integrality on these variables can also be relaxed.

3.4 Case Study

All computational tests were run on an IBM RS 6000/370 using CPLEX (version 3.0) [Bixby 1994]. All run-times shown in the tables are in seconds. We measure the difference between two solution values by calculating the difference between the pay-and-credit of the two solutions.

3.4.1 Data

We conduct our experiments using actual data provided by a long-haul U.S. airline. Table 3.1 shows five test problems, named $P1$ through $P5$. Problems $P1$ through $P4$ have two crew bases while $P5$ has only one crewbase. *Flights* denotes the number of operational flights. *Deadheads* refers to the number of deadheads that were picked by the deadhead selection procedure developed by Barnhart, et al [1995] (Figure 3-2). *Duties* refers to the total number of duties.

3.4.2 Solution of CPP - Baseline for Comparison

To provide a baseline against which we can compare the performance of our approach, we solved the traditional CPP model using branch-and-bound, generating columns

Problem	P1	P2	P3	P4	P5
Crew Bases	2	2	2	2	1
Flights	832	825	633	875	815
Deadheads	2217	909	785	1765	1655
Duties	4094	2802	2229	4226	4288

Table 3.1: Test Problems' Characteristics

only at the root node of the branch-and-bound tree[4]. In general, we restrict the branch-and-bound tree to a search of 500 nodes. However, this failed to find any IP solution for problems $P2$ and $P4$, and so we raised the node limit to 1000.

We ran a modified version of Baker's heuristic [Baker 1981b] at each iteration of column generation to get an upper bound on the optimal solution value. Baker's heuristic selects columns to be included in a feasible solution based on the lowest ratio of pairing cost to number of uncovered flight legs that the pairing will cover. We modified this heuristic by greedily choosing pairings based on the ranking

$$\min_j \frac{(1 - w_j^*)c_j}{\bar{l}_j},$$

where w_j^* is the optimal LP value of pairing j in the current column generation iteration, c_j is the cost of column j and $\bar{l}_j$ is the number of flight legs in pairing j that are not yet covered by the heuristic. The rationale of our modified ranking rule is that it will prefer low cost pairings, pairings favored by the LP solver (i.e., pairings whose solution values are closer to unity), and pairings that cover more uncovered flights in the schedule. We, therefore, use the LP solution as a *guide* in generating a feasible integer solution.

Table 3.2 shows that using the method given in Figure 3-1 (allowing use of a pre-selected set of deadhead flights), the average time to solve CPP is about 2 hours, 20 minutes, and the average difference in pay-and-credit between the optimal LP

[4]The root node LP is solved using the primal steepest edge option in CPLEX. The branch-and-bound tree is solved using the following options: *setadvind*(1), *setpreind*(1), *setaggind*(1), *setcoeredind*(1), *setcliques*(1), *setcovers*(1), and *setheuristic*(1).

Problem	P1	P2	P3	P4	P5	Average
Pay-and-Credit						
CPP-LP(A)	84.78%	91.51%	136.80%	56.64%	143.20%	
CPP-IP(B)	84.88%	91.68%	136.81%	56.94%	143.45%	
Solution Times						
Time to CPP-LP	6649	2872	678	10593	2850	**4728**
Time to CPP-IP	2594	861	23	14521	260	
Total Time	9243	3733	701	25114	3110	**8380**
Optimality Gap						
(B)-(A)	0.10%	0.17%	0.01%	0.30%	0.25%	**0.17%**

Table 3.2: Solution of CPP - Baseline for Comparison

solution and the best integer solution is small (about 0.17%). This suggests that we can generate near optimal integer solutions by generating columns only at the root node, and by limiting the branch and bound search tree.

3.4.3 Improving Solutions with Deadhead Selection

Deadheads are very crucial to improving the solution value in long-haul problems [Barnhart, et al 1995]. The problem, however, is that there are too many possible deadheads (denoted *Potential Deadheads*[5] in Table 3.3.) As a first step, we follow Barnhart et al.[1995] and eliminate all flights that start or end at airports where no operational flights start or end. This gives us a smaller set of flights (denoted *Elim 1* in Table 3.3). To further reduce the number of deadheads, we assume that it is not efficient for a crew ready for work to wait extensively on the ground before deadheading. Using this, we eliminate all deadhead flights that do not start within a specified time, say two hours, of when some crew is ready at an airport. This gives us a manageable number of deadheads (*Elim 2* in Table 3.3). With this approach, we reduce the number of possible deadheads from a few million to tens of thousands. We then use this reduced set of flights as the input to the deadhead selection procedure.

The overall deadhead selection process is shown in Figure 3-3. At each iteration,

[5]From the Official Airlines Guide[1991].

Problem	P1	P2	P3	P4	P5	Average
Potential Dhds	3084018	3084018	2643444	2643444	2643444	
Elim. 1	145922	145922	125076	125076	255594	
Elim. 2	21616	13190	9769	21315	33089	
Dhds Selected	310	390	296	193	429	**324**
Time taken	703	399	229	626	327	**457**

Table 3.3: DPP and Deadhead Selection

we solve the LP relaxation of DPP (called DPP-LP)[6]. Then, as in Barnhart, et al [1995], we use the DPP-LP dual solution to select additional deadheads that are included in DPP-LP in the next iteration. These iterations are repeated until the improvements in the DPP-LP solution value are negligible. In our experiments, we observed that five iterations were usually enough.

At each iteration, any deadhead that has a non-zero LP solution value in DPP is added to the set of selected deadheads input to the crew pairing solver. Table 3.3 shows that on an average, our deadhead selection and approximate crew pairing solution selected about 324 deadheads and took an average of about 8 minutes. In order to compare this to the method of Barnhart et al.[1995], we note that their method will take at least 1 hour and 18 minutes - the average time taken to solve *one iteration* of CPP-LP (Table 3.2). Therefore, significant speed-up can be achieved by only approximately solving the crew pairing LP.

To evaluate the merit of our approach, we add the selected deadheads to the CPP input and solve CPP using the conventional approach (Figure 3-1). Table 3.4 shows that LP-IP gaps in pay-and-credit are still small (0.23%), but due to greater input size, the problems take longer to solve (about 2 hours and 38 minutes to solve plus the time to generate deadheads, yields a total of about 2 hours and 45 minutes). However, the important point is that the deadheads improves the solutions considerably. Table 3.5 shows that the maximum difference in pay-and-credit between the baseline case solutions and these is about 40% and the average difference is about 20%. Barnhart,

[6]We solve DPP-LP using dual simplex with the CPLEX's *netopt* option.

Problem	P1	P2	P3	P4	P5	Average
Pay-and-Credit						
CPP-LP(C)	78.56%	78.31%	101.13%	49.82%	103.45%	
CPP-IP(D)	78.58%	78.45%	101.13%	50.22%	104.02%	
Solution Times						
Time to CPP-LP	4921	2073	935	10619	3464	
Time to CPP-IP	315	132	58	24023	693	
Total Time	5236	2205	993	34642	4157	**9447**
Optimality Gap						
(D)-(C)	0.02%	0.14%	0.00%	0.40%	0.57%	**0.23%**

Table 3.4: Effects of DPP and Deadhead Selection

Problem	P1	P2	P3	P4	P5	Average
Pay-and-Credit						
CPP-IP(B)	84.88%	91.68%	136.81%	56.94%	143.45%	
Pay-and-Credit						
CPP-IP(D)	78.58%	78.45%	101.13%	50.22%	104.02%	
Improvement						
(B)-(D)	6.30%	13.23%	35.68%	6.72%	39.43%	**20.27%**

Table 3.5: Baseline CPP vs. New CPP Solution

et al [1995] state that for the datasets under consideration, each percentage point reduction in pay-and-credit amounts to savings of about a quarter million dollars a year. Our results, therefore, translate to annual savings of $5 to $10 million.

3.4.4 Speeding up CPP

Given the vastly improved solutions generated when we combined our approximate solution approach with the conventional CPP solution method, we focused on how to speed up overall solution time. We identified two strategies: first, we used our DPP solution to *guide* column generation and second, we used bounding techniques to *speed-up* column generation.

Problem	P1	P2	P3	P4	P5	Average
Pay-and-Credit						
CPP-LP(C)	78.56%	78.31%	101.13%	49.82%	103.45%	
CPP-IP(E)	78.58%	78.46%	101.13%	49.82%	104.26%	
Solution Times						
Time to CPP-LP	3078	1169	558	3805	2338	
Time to CPP-IP	331	68	25	263	871	
Total Time	3409	1237	583	4068	3209	**2501**
Optimality Gap						
(E)-(C)	0.02%	0.15%	0.00%	0.00%	0.81%	**0.20%**

Table 3.6: Speed Up - CPP Solution with Restricted Column Generation

Restricted Column Generation

In our baseline case, the LP relaxation of CPP is solved using column generation, with pairings generated using the entire set of duties. Since the dual variables generated by the master problem in the early iterations may be far from optimal, many pairings may be generated that will not be used in an optimal solution to CPP and the resulting solution time is increased unnecessarily.

To avoid this, we use a two-step column generation method to solve CPP-LP. In the first step, columns are generated using only those duties in the optimal DPP-LP solution. Once no additional pairings are generated in this fashion, we move to the second step of the process in which we generate columns using *all* duties until CPP-LP is solved.

Table 3.6 shows that the average total time taken to solve CPP (i.e., about 42 minutes plus time to select deadheads giving a total of about 49 minutes) using this two-step process was a 65% improvement over the 2 hours and 20 minutes (Table 3.2) taken to solve CPP with the baseline approach.

Problem	P1	P2	P3	P4	P5	Average
Pay & Credit for CPP-LP(C)	78.56%	78.31%	101.13%	49.82%	103.45%	
Time to CPP-LP	3078	1169	558	3805	2338	**2190**
Pay & Credit for DPP-LP(F)	78.41%	77.51%	99.70%	49.82%	101.57%	
Time to DPP-LP	60	21	10	43	3	**27**
(C)-(F)	0.15%	0.80%	1.43%	0.00%	1.88%	**0.85%**
Pay & Credit for DPP-IP(G)	78.47%	77.74%	100.44%	50.00%	102.42%	
Time to DPP-IP	708	300	97	600	18	**345**
(C)-(G)	0.09%	0.57%	0.69%	-0.18%	1.03%	**0.44%**

Table 3.7: Bounding with DPP-LP and DPP-IP

Lower Bound Generation

Since every solution of CPP is feasible to DPP but not vice versa, DPP-LP is a lower bound to CPP-LP, and DPP-IP[7] is a lower bound to CPP-IP. In Table 3.7 we present the DPP bounds. In our evaluation of these bounds, we make two observations. First, the DPP bounds are obtained quickly. DPP-IP can be solved using one order of magnitude lesser time than CPP-LP, while DPP-LP can be solved using two orders of magnitude lesser time than CPP-LP. Second, the DPP bounds are tight. Pay-and-Credit for both DPP-LP and DPP-IP are on average within 1% of the optimal pay-and-credit for CPP-LP. In one case ($P4$), the DPP-IP optimal solution is actually greater than the optimal CPP-LP solution value, providing a stronger bound in a fraction of the time.

We use the lower bounds generated in solving DPP-LP to reduce overall computation time as follows. We multiply the DPP-LP solution value by a factor (e.g., 1.01), and we terminate solution of the root node CPP-LP when its solution value dips below this number. Table 3.8 shows that by using this termination criterion, we eliminate some of the tailing time of column generation. In fact, the average time to

[7]DPP-IP is solved using the same CPLEX options as CPP-IP, except that the branch-and-bound search is extended to 1000 nodes.

Problem	P1	P2	P3	P4	P5	Average
Pay-and-Credit						
CPP-LP(H)[8]	79.73%	79.12%	101.69%	51.15%	103.49%	
CPP-IP(I)	79.74%	79.28%	101.71%	51.57%	104.26%	
Solution Times						
Time to CPP-LP[8]	877	721	375	236	2154	
Time to CPP-IP	74	75	32	127	723	
Total Time	951	796	407	363	2877	**1079**
Optimality Gap						
(I)-(E)	1.16%	0.82%	0.58%	1.75%	0.00%	**0.86%**

[8]Not run to LP optimality - terminated by bound = DPP-LP*1.01

Table 3.8: Final CPP Solution

solve CPP compared to the baseline case was reduced by over 87%, to 18 minutes. Since deadhead selection takes an average of 8 minutes, the average total time to generate a near-optimal solution for CPP is about 26 minutes - an 82% improvement in the total solution time of the baseline case.

An important observation is that solution times are reduced significantly and solution quality is adversely affected only to a small extent. To show this, we calculate the difference in pay-and-credit values of the best solution generated with and without premature termination, i.e., termination using DPP-LP bounds. Table 3.8 shows that the average difference is only 0.86%

The DPP-LP lower bound can also be used in branch-and-price. At each node of the branch-and-price tree, we have two options. If the DPP-LP bound is higher than the current best solution, then the node can be fathomed. If the DPP-LP bound is lower, it can be used in place of the CPP-LP optimal solution value, thereby allowing early termination of column generation and eliminating some of the tailing effects. This speed-up can be exploited by planners in constructing the schedule. For many alternate schedules, the best possible solution value for the flight schedule can be generated quickly by solving DPP-LP. Existing CPP-LP column generation solution techniques are limited in this capability because they require excessive computation time to determine good estimates of the best possible solution values.

3.4.5 Scenario Analysis

Consider the following simple heuristic for obtaining feasible solutions to CPP:

1. Solve DPP-IP to optimality.

2. Partition the DPP-IP solution into a set of (possibly *illegal*) pairings. This is always possible since it corresponds to decomposing arc flows into path flows [Ahuja, et al 1993]. All pairings may not be legal because the maximum time-away-from-base constraint may be violated for one or more pairings.

3. Transform illegal pairings into legal pairings. For each pairing violating time-away-from-base constraints, break the pairing into one or more legal pairings by adding deadheads at the start (and/or end) that depart from (return to) a crewbase and connect up with the pairing.

We use this simple heuristic to perform "what-if" scenario analyses. For example, in problem $P2$, we assume that the two crew bases are cities A and B. However, in Table 3.9 we observe that 32% of all operational flights begin or end at another city, say C.

We, therefore, investigate and gauge the benefits of adding a new crew base in city C. Rather than solve CPP, since it is computationally expensive, we build a feasible solution for DPP-IP using our heuristic. This process takes about 20 minutes and reduces pay-and-credit from 91.10% using only two crewbases to 78.71%, with an estimated savings of $ 3 million annually. Since we generate a feasible, and not optimal solution, we are quickly able to get a *ball-park* estimate of the impact of the proposed change.

Description	Value
Problem	P2
Total Flights	1734
Flights thru new Base	483
Ratio (base flt/total)	28%
Total Oper. Flt.	825
Oper. Flt. thru base	267
Ratio (base oper/total)	32%
Without New Base	
Pay-and-Credit for DPP-IP(J)	91.10%
Time to DPP-IP	285
With New Base	
Pay-and-Credit for DPP-IP(K)	73.32%
Pay-and-Credit for DPP-Heur(L)	78.71%
Improvement = (J)-(L)	12.39%
Total Time	1238

Table 3.9: Scenario Analysis

Chapter 4

Aircraft Scheduling

The chapter studies the *fleet assignment* problem, the *through flight assignment* problem, and the *aircraft maintenance routing* problem, for long-haul schedules. Figure 4-1 shows the sequence in which these problems are solved. In this chapter, we present each of these problems and detail the modeling and solution issues. We then present the long-haul *aircraft scheduling* problem, i.e., the problem that combines the above decisions, and shows how these three decision modules can be solved simultaneously using one model.

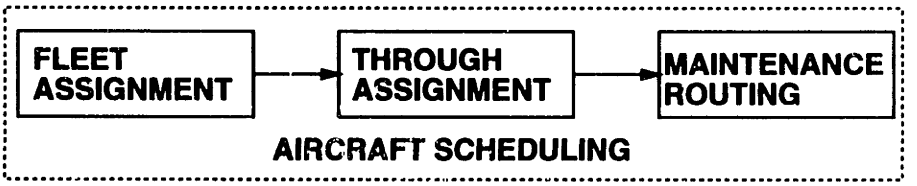

Figure 4-1: The Aircraft Scheduling Problem

4.1 Introduction

After creating a flight schedule, the airline must decide what equipment type, or *fleet type*, should be assigned to each flight. This is referred to as the *fleet assignment* problem [Abara 1989]. The airline must then solve the *through flight assignment*

problem which decides which sequences of flights should be assigned the same aircraft to maximize *through revenues* [Talluri and Gopalan 1995]. Through revenues refer to the extra revenues generated by providing itineraries which do not require a change of aircraft. After that, the airline must decide which specific aircraft, or *tail number*, of each equipment type should fly each flight leg or sequence of through flights so each aircraft visits maintenance stations on a regular basis. Routing the aircraft to satisfy through assignments and maintenance criterion is called *aircraft maintenance routing* [Clarke et al 1995].

Due to the high frequency and density of domestic flight networks (section 1.2.3), domestic schedules are associated with several possible assignments of aircrafts, or *tail numbers*, to flights. The point-to-point long-haul networks, on the other hand, are characterized by sparse activity at stations, yielding few possible aircraft assignments. Therefore, in domestic problems, a sequential process of solving the fleet assignment, through assignment, and maintenance routing problems will allow the generation of aircraft assignments that satisfy all Federal Aviation Administration (FAA) maintenance criterion [Subramanian et al 1994].

In a long-haul schedule however, conventional wisdom indicates that a sequential solution of these problems may not yield a legitimate maintenance schedule. This chapter validates this hypothesis and describes models and methods for the solution of the long-haul *aircraft scheduling* problem, i.e., the combined long-haul fleet assignment, through flight assignment, and aircraft maintenance routing problem. Although the models described in this chapter are tailored to solve long-haul problems, they are applicable, but not necessarily crucial, to domestic problems.

4.1.1 Contributions

This chapter makes the following contributions:

- In section 4.2, we present a long-haul fleet assignment model that is enhanced using maintenance considerations.

- In sections 4.3 and 4.4, we present the through flight assignment problem and the aircraft maintenance routing problem and related issues.

- In section 4.5, we present an aircraft scheduling model that integrates the fleet assignment problem, the through flight assignment problem, and the aircraft maintenance routing problem. We also show that this model is a generalization of the aircraft maintenance routing model. Finally, to solve the aircraft scheduling problem, we develop specialized branch-and-price solution methods.

- In section 4.7, we use data from a long-haul airline to show that the sequential solution of fleet assignment and aircraft maintenance routing does not always yield feasible solutions. We then go on to show how near-optimal solutions can be obtained for the integrated problem in reasonable amounts of time.

- Finally, in section 4.8, we extend the aircraft scheduling model to include maintenance base planning considerations, and show how it can be used to decide which maintenance stations to open or close, which fleets to maintain at each station, and how many maintenance workers should be staffed at each station on each day of the week.

4.2 Fleet Assignment

The goal of the fleet assignment problem is to generate an assignment of fleets to flights, that satisfies certain constraints [Abara 1989].

Fleet Assignment Input

The main input for the long-haul fleet assignment problem is a weekly schedule of flight legs. A *flight leg* is defined as a non-stop flight between two cities, and a *weekly* schedule is one in which the flight legs repeat identically each week. For example, if a flight in a weekly schedule is scheduled on a Monday and Wednesday, it is assumed that it is scheduled on *every* Monday and Wednesday.

The other inputs to this problem are a set of aircraft of various equipment types, called *fleets*, the cost of flying each flight leg, by fleet type, and *turn-time* restrictions at each station, by fleet type. Turn-time refers to the time taken by the ground crew to ready the plane and turn it around for the next flight. Turn time requirements can vary by station and fleet type. Turn times are generally higher at busier airports and for larger aircraft [Clarke et al 1995b].

Fleet Assignment Output

The output of the fleet assignment model is an assignment of fleets to flights that either maximizes revenues or profits, or minimizes costs. In addition, the assignment has to satisfy the following requirements [Hane et al 1994].

- **Flight Coverage:** Each flight must be assigned to *exactly* one fleet. In some cases, this requirement can be relaxed to enforce that each flight be covered *at most* once. For example, a schedule planner may have an extra list of flight legs that need not be covered but can be used if necessary [Desaulniers et al 1994b]. By requiring the model to cover the flight at most once, some flights can be rejected.

- **Aircraft Balance:** For each fleet, the number of aircraft flying into an airport must equal the number flying out of that airport. This requirement ensures a continuity of aircraft at each airport at each point in time during the week.

- **Fleet Size:** For each fleet, the number of aircraft used must not exceed the number available.

4.2.1 The Cost of Assigning a Flight to a Fleet

The cost of flying a flight using a specific fleet is the sum of the aircraft operating costs, and the opportunity costs of losing *spilled* passengers due to excess demand. Some spilled passengers may take an alternative flight on the same airline and will hence be *recaptured*. Spill costs are adjusted by the revenues from these recaptured

passengers. Since spill costs are a function of both demand for a flight and aircraft capacity, total spill cost varies for each fleet-flight pair. The cost of a fleet-flight pair is computed by adding spill and operating costs [Subramanian et al 1994].

While fuel costs and crew costs of a fleet-flight leg pair may be straightforward to estimate, other costs involving the introduction of a new fleet to a station may be difficult. The associated costs include maintenance crew training or relocation and spare parts inventory. Also challenging is the estimation of spill and recapture rates [Belobaba 1987]. In addition, dependence between flight legs make the exact estimation of their costs nearly impossible. The problem is that many passengers have more than one flight leg in their itinerary. Estimating spill costs for fleet-flight pairs cannot capture the fact that a passenger spilled from one leg of his/her itinerary is spilled from all other legs also [Farkas 1996].

4.2.2 Earlier Work

The fleet assignment problem has been well studied [Abara 1989, Daskin and Panayotopoulos 1989, Berge and Hopperstad 1993, Dillon et al 1993, Subramanian et al 1994, Hane et al 1994, Gu et al 1994, Desaulniers et al 1994b, 1995b, Clarke et al 1995b]. In this section, we develop a model for long-haul fleet assignment by extending the work of Hane et al [1994].

4.2.3 Flight Networks

The core of any fleet assignment model is a flight network. In our fleet assignment model we use a time-line network of flights (section 2.3.1) [Hane et al 1994]. Since the fleet assignment problem tries to maintain balance for each fleet, one separate network is created for each fleet type. Each node is an arrival or a departure, and each arc is either a flight arc or a ground arc. Specifically, each flight network has three types of arcs - *flight arcs*, *maintenance arcs*, and *ground arcs*.

1. One *flight arc*, and its start and end nodes, is created for each flight in the schedule, in each fleet network. The location and time of the starting node is

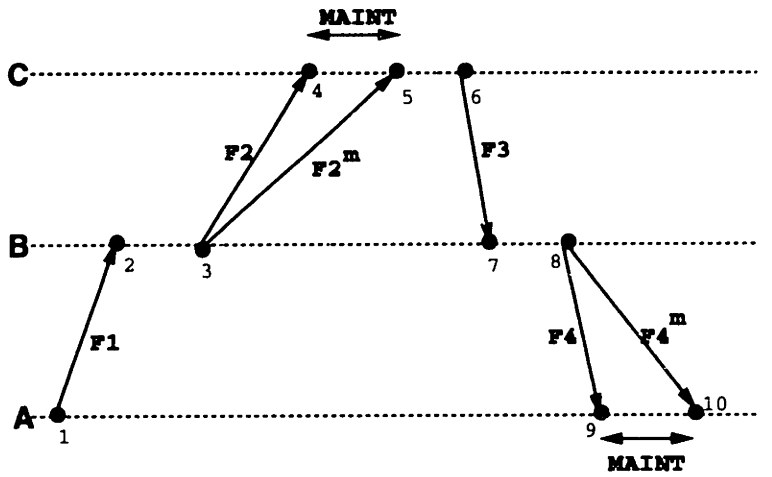

Figure 4-2: Flight Arcs and Maintenance Arcs

the origin and starting time of the corresponding flight. The location and time of the ending node is the destination and *ready time* of the corresponding flight. The ready time of the flight is defined as the sum of the arrival time and the turn-time, i.e., the time needed to turn the aircraft flying that flight so that it is ready for the next flight.

2. In each fleet network, one *maintenance arc*, and the corresponding start and end nodes, is created for each flight that ends at a maintenance station for that fleet. Maintenance arcs are sometimes called *split arcs* [Clarke et al 1995b]. The starting node is identical to that of the corresponding flight arc. The location and time of the ending node is the destination and *maintenance ready time* of the corresponding flight, i.e., the time that the aircraft flying that flight is ready after it has been serviced with an "A" check (this will be discussed in greater detail in section 4.4). By creating a separate fleet network for each fleet type, we allow the *ready time* and *maintenance ready time* to vary by fleet.

Example 6 *Figure 4-2 shows an example of a time-line flight network for one fleet type, three cities,* **A**, **B**, *and* **C**, *and four flights,* **F1**, **F2**, **F3**, *and* **F4**. **A** *and* **C** *are maintenance stations, and so we create maintenance arcs for flights* **F2** *and* **F4**, *called* $\mathbf{F2^m}$ *and* $\mathbf{F4^m}$ *respectively.*

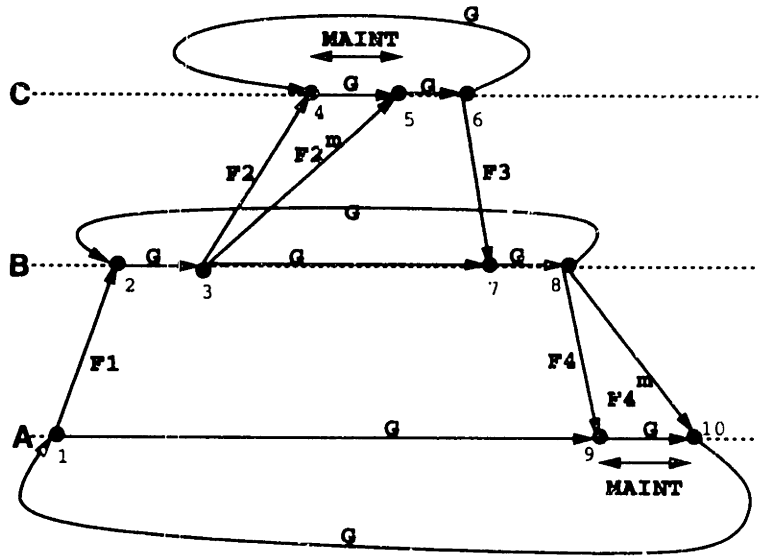

Figure 4-3: Ground Arcs and Wrap Around Arcs

3. All the starting and ending nodes are sorted chronologically at each station, and *ground arcs* are placed between successive nodes. A special type of ground arc, a *wrap-around arc*, provides continuity between the last event at a station and the first event at that station. While these arcs represent aircraft overnighting on the last day of the week, they enable the modeling of the fleet assignment problem as a circulation problem and eliminate the need to insert supplies and demands into the network.

Example 7 *In figure 4-3, ground arcs, denoted* **G**, *are placed between chronologically successive nodes at each city. Each station has a wrap-around arc.*

In order to count the aircraft used by the model, a time is arbitrarily chosen in the flight network as the *count time*. The total number of aircraft of each fleet type assigned to each arc is counted. Since we solve a long-haul problem in which the schedule is weekly, the length of the time-line networks is 7 days. The domestic fleet assignment problem on the other hand is a daily problem, due to which the length of the time-line networks used to solve such problems is 24 hours [Hane et al 1994].

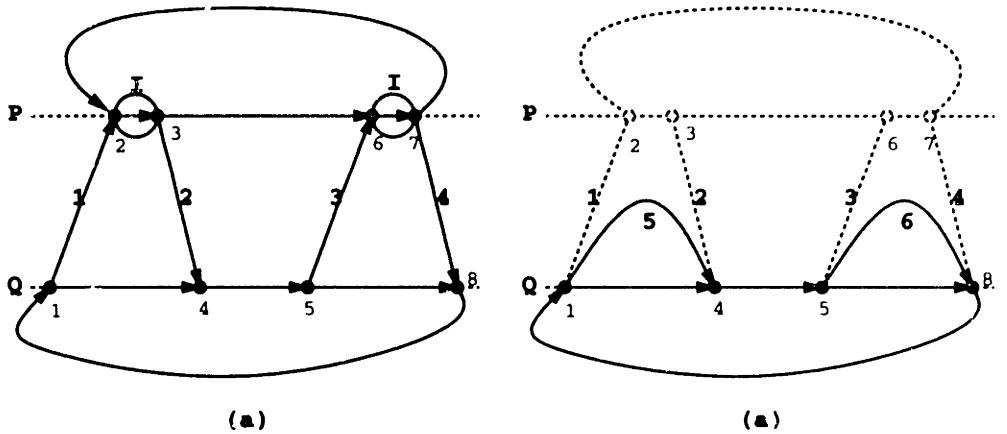

Figure 4-4: Island Formation

Network Simplification

Node aggregation is a method of simplifying time-line networks by clumping together a set of arrivals followed by a set of departures (section 2.3.1) [Hane et al 1994]. Another concept, that is especially applicable to fleet assignment problems, is that of *islands* [Hane et al 1994]. This is based on the observation that in such problems, if the utilization of aircraft is heavy, then at some stations, periods of time can be identified where there will be no aircraft on the ground. The ground arcs corresponding to those periods can be eliminated to create *islands* of activity with no aircraft on the ground between the islands [Hane et al 1994].

Example 8 *Figure 4-4(a) shows two cities* **P** *and* **Q**, *and four flights,* **1** *through* **4**, *that can be flown by one aircraft. If only one aircraft is available, then the ground arc* $\{\mathbf{3},\mathbf{6}\}$ *and the wrap-around arc* $\{\mathbf{7},\mathbf{2}\}$ *at station* **P** *will have no aircraft. If we eliminate these two arcs, then we have two islands, each denoted by* **I**. *These islands can be consolidated to eliminate two nodes and two ground arcs. Finally, these islands are unique in that each has one arrival followed by one departure, in which case the flights can be consolidated as well. Flight 1 (3) will be assigned the same aircraft type as flight 2 (4), otherwise conservation of flow of aircraft of each fleet will not be achieved. Flights* **1** *and* **2** *can then be consolidated into flight* **5**, *and flights* **3** *and* **4** *can be consolidated into* **6**. *This not only eliminates two nodes, it also reduces the*

number of decision variables. In this example, 4 nodes, 4 ground arcs, and 2 flight arcs are eliminated.

Islands can also be used to limit the number of arcs in a flight connection network (section 2.3.1). To check if a connection between a pair of flights should be allowed at a station for a given fleet, we check the time-line network to see if any ground arc has been eliminated by islands, between the arrival of the first flight and the departure of the second, for that fleet. If no ground arc has been eliminated, then the connection is valid. Otherwise, the connection is disallowed [Barnhart 1996].

4.2.4 The Fleet Assignment Model

The fleet assignment model of Hane et al [1994] is presented below. This model is based on the time-line networks described above and has been modified for the long-haul problem. The notation for this model is given in Table 4.1.

The Basic Fleet Assignment Model

The basic fleet assignment model is given below [Equations (4.1)-(4.6)].

$$\min \sum_{k \in K} \sum_{i \in F} e_{ik} z_{ik} \tag{4.1}$$

$s.t.$

$$\sum_{k \in K} \sum_{i \in F} \delta_{pik} z_{ik} = 1 \quad \forall p \in F \tag{4.2}$$

$$\sum_{i \in F} b3_{nik} z_{ik} + \sum_{h \in H_k} b4_{nhk} w_{hk} = 0 \quad \forall n \in M_k, \forall k \in K \tag{4.3}$$

$$\sum_{i \in F} d3_{ik} z_{ik} + \sum_{h \in H_k} d4_{hk} w_{hk} \leq N_k \quad \forall k \in K \tag{4.4}$$

$$z_{ik} \in \{0,1\} \; \forall i \in F, \forall k \in K \tag{4.5}$$

$$w_{hk} \geq 0 \quad \forall h \in H_k, \forall k \in K \tag{4.6}$$

The objective function (4.1) minimizes the cost of the assignment of fleets to flights. The cost e_{ik} of assigning flight i to fleet k is discussed in section 4.2.1.

F	=	Set of Flights
K	=	Set of Fleets
N_k	=	Number of aircraft in fleet k
H_k	=	Set of all ground arcs in fleet k
M_k	=	Set of all nodes for fleet k
e_{ik}	=	Operating cost of flight i flown by fleet k
z_{ik}	=	1, if flight arc i is used by fleet k, 0 otherwise
z^m_{ik}	=	1, if maintenance flight arc i is used by fleet k,0 otherwise
δ_{pqk}	=	1, if $p = q$, 0 otherwise
δ^m_{pqk}	=	1, if $p = q$ and maintenance arc p exists for fleet k, 0 otherwise
w_{hk}	=	Number of aircraft on ground arc h in fleet k
$b3_{nik}$	=	1, if flight arc i in fleet k begins at node n,
	=	-1, if flight arc i in fleet k ends at node n, 0 otherwise
$b3^m_{nik}$	=	1, if maintenance arc i in fleet k begins at node n,
	=	-1, if maintenance arc i in fleet k ends at node n, 0 otherwise
$b4_{nhk}$	=	1, if ground arc h in fleet k begins at node n,
	=	-1, if ground arc h in fleet k ends at node n, 0 otherwise
$d3_{ik}$	=	1, if flight arc i in fleet k crosses the count line, 0 otherwise
$d3^m_{ik}$	=	1, if maintenance arc i in fleet k crosses the count line, 0 otherwise
$d4_{hk}$	=	1, if ground arc h in fleet k crosses the count line, 0 otherwise
R^Q_t	=	Set of flights that begin on any of Q consecutive days starting on day t

Table 4.1: Notation for the Fleet Assignment Model

Equations (4.2) ensure that each flight is assigned to exactly one fleet. Equations (4.3) maintain the balance of aircraft at each station for each fleet at each instance of time. Equations (4.4) make sure that the model does not use more aircraft than available. Equations (4.5) and (4.6) force the integrality of the flight arcs and ground arcs. The integrality of the flight arc variables will enforce the integrality of the ground arc variables.

The Enhanced Fleet Assignment Model

While the solution to the fleet assignment model is the least cost feasible assignment, it will not be useful if *maintenance feasible* rotations cannot be generated from it. A rotation is a sequence of flights that starts and ends at the same location. A maintenance feasible rotation is one that allows the aircraft to be serviced with an "A" check once every four days[1].

We attempt to enhance the fleet assignment model by enforcing a certain minimum number of maintenance *opportunities*. This is done by forcing a minimum flow on the maintenance arcs, using a method similar to that of Hane et al[1994] and Clarke et al [1995b]. They solve the daily domestic fleet assignment problem, and aircraft must visit a maintenance station once every four days. To provide *enough* maintenance opportunities, they force the total flow on the maintenance arcs to be not less than $\lceil N_k/4 \rceil$, where N_k is the number of aircraft in fleet k.

While this constraint form is essential and accurate for the daily fleet assignment problems they solve (since the operations on each day should be identical), this constraint form becomes restrictive in a weekly problem where the operations need repeat only on a weekly basis.

Example 9 *Consider a weekly schedule that can be flown by only one aircraft, i.e,* $\mathbf{N_k = 1}$. *Given a limit of 4 days between maintenance stops, a feasible maintenance*

[1]The issues of maintenance are presented in detail during the discussion of the aircraft maintenance routing problem (section 4.4). For now, it will suffice to say that a maintenance feasible rotation is one that can be serviced with an "A" check once every four days, where an "A" check is one of the maintenance service checks mandated by the FAA [Clarke et al 1995].

schedule can be obtained if the aircraft is maintained, say, every Thursday and Sunday, satisfying the four day requirement. However, a constraint requiring the flow on the maintenance arcs on each day to be greater than $\lceil \mathbf{N_k}/\mathbf{4} \rceil = \lceil \mathbf{1}/\mathbf{4} \rceil$ *would force one maintenance stop on* **each** *day, which is excessive.*

In our problem, even though maintenance is required every four days, we adopt a more aggregate approach. We force a minimum of N_k maintenance opportunities in every four day interval, and at least $\lceil 7N_k/4 \rceil$ maintenance opportunities over the week, for fleet k. Note that these constraints do not guarantee that each aircraft will actually be maintained every four days. The notation for this model is the same as for the basic model and the model is given in Equations (4.7)-(4.14).

$$\min \sum_{k \in K} \sum_{i \in F} e_{ik} z_{ik} + \sum_{k \in K} \sum_{i \in F} e_{ik} z^m_{ik} \tag{4.7}$$

$s.t.$

$$\sum_{k \in K} \sum_{i \in F} \delta_{pik} z_{ik} + \sum_{k \in K} \sum_{i \in F} \delta^m_{pik} z^m_{ik} = 1 \quad \forall p \in F \tag{4.8}$$

$$\sum_{i \in F} b3_{nik} z_{ik} + \sum_{i \in F} b3^m_{nik} z^m_{ik} + \sum_{h \in H_k} b4_{nhk} w_{hk} = 0 \quad \forall n \in M_k, \forall k \in K \tag{4.9}$$

$$\sum_{i \in F} d3_{ik} z_{ik} + \sum_{i \in F} d3^m_{ik} z^m_{ik} + \sum_{h \in H_k} d4_{hk} w_{hk} \leq N_k \quad \forall k \in K \tag{4.10}$$

$$\sum_{i \in R^a_t} z^m_{ik} \geq N_k \quad \forall t, \forall k \in K \tag{4.11}$$

$$\sum_{i \in F} z^m_{ik} \geq \left\lceil \frac{7N_k}{4} \right\rceil \forall k \in K \tag{4.12}$$

$$z_{ik}, z^m_{ik} \in \{0, 1\} \; \forall i \in F, \forall k \in K \tag{4.13}$$

$$w_{hk} \geq 0 \quad \forall h \in H_k, \forall k \in K \tag{4.14}$$

The objective function (4.7) minimizes the total cost of assigning flights to fleets. The costs of assigning a flight to a fleet is discussed in section 4.2.1. Equations (4.8) ensure that each flight is assigned to exactly one fleet. Equations (4.9), one set for each fleet, ensure that flow balance is maintained between the flight arcs, maintenance arcs, and ground arcs. Equations (4.10), one for each fleet, ensure that the number of aircraft used do not exceed the number available. Equations (4.11) and (4.12)

strengthen the lower bound and ensure that a minimum flow is maintained on the maintenance arcs. Equations (4.13) and (4.14) ensure the integrality of the flow in the network.

4.3 Through Flight Assignment

The through flight assignment problem tries to select sequences of flights to be flown by the same aircraft to maximize revenues [Talluri and Gopalan 1995].

4.3.1 Motivation

Through itineraries, i.e., those in which passengers do not have to change aircraft, generate extra revenues over *connects*, i.e., itineraries in which the passengers change planes, because passengers are willing to pay a premium to stay on the same aircraft rather than make a tight connection where the chances of lost luggage is also greater [Talluri and Gopalan 1995]. The through flight assignment problem tries to construct a set of itineraries that maximizes through revenues.

4.3.2 Earlier Work

Since through revenues are represented by a complex nonlinear relationship, the modeling of this problem can be quite involved. Therefore, most earlier work has relied on heuristics [Hersh and Ladany 1978, Bard and Cunningham 1987]. Talluri and Gopalan [1995] develop a through assignment model as a simple set packing problem. They calculate the increase in revenues from a through assignment using the traffic forecasts for the market, the average revenue of passengers in that market, and the *desirability* of that through as compared to the desirabilities of other itineraries serving the same market. The desirability of a through depends on a number of factors such as number of stops, circuity of the through, etc.

In domestic hub-and-spoke problems (for reasons given in section 1.2.5), the generation of maintenance feasible rotations is usually possible even with *a priori* through

flight assignments [Subramanian 1994]. However, in the long-haul problems we consider, generation of maintenance feasible rotations may not be possible even if the through assignments have not been fixed *a priori*. Finding maintenance feasible rotations, given *a priori* fleeting *and* through assignment decisions is even less likely.

4.4 Aircraft Maintenance Routing

Given the set of flights assigned to a fleet, *the aircraft maintenance routing problem tries to find rotations* (defined earlier) *that collectively cover each flight exactly once and satisfy aircraft maintenance requirements* [Clarke et al 1995].

4.4.1 Problem Definition

The fleet assignment solution assigns fleet types, not specific planes, to flight legs. The actual assignment of specific aircraft, or *tail numbers*, to the fleeted schedule so that through assignments (selected *a priori*) are satisfied and each aircraft visits maintenance stations on a regular basis is called the *aircraft maintenance routing* problem [Clarke et al 1995]. For a given fleet assignment and a given set of through assignments, this problem is to determine a set of aircraft rotations that satisfy the following requirements [Clarke et al 1995]:

- **Coverage:** Each flight assigned to a fleet is flown by exactly one aircraft in that fleet.
- **Aircraft Count:** The total number of aircraft used by the rotations does not exceed the number of available aircraft of that equipment type.
- **Maintenance Regulations:** Each rotation must satisfy the maintenance requirements of the FAA.

Type of Check	Time between Checks	Required Labor
Periodic Service	Daily (overnight or during downtime during flight day)	2 man-hours
"A" Checks	About 60 flight hours (about once a week)	10-20 man-hours
"B" Checks		
Narrowbody "B"	About 300-500 flight hours (about once a month)	100 man-hours
Widebody "B"	About 300-500 flight hours (about once a month)	200-300 man-hours
"C" Checks		
Narrowbody "C"	"Light C" Checks are done once every year	2,100 man-hours 3 days to complete
	"Heavy C" Checks are done once every 4 "Light" C's	20,000-30,000 man-hours 3-5 weeks to complete
Widebody "C"	All are "Heavy C" and are done every 15-18 months	10,000 man-hours 2 weeks to complete

Table 4.2: Various Types of Maintenance Performed by American Airlines

4.4.2 Maintenance Requirements

The FAA mandates four main categories of safety checks on airlines. These are called "A", "B", "C", and "D", and vary in scope, duration, and frequency [Clarke et al 1995]. Table 4.2 shows the various kinds of maintenance checks carried out by American Airlines[2]. The maintenance checks that we model in the routing problem are the "A" checks. As mandated by the FAA, each aircraft flown by an airline has to be subjected to an "A" check for every 65 hours of flying. Airlines typically have more stringent maintenance requirements and have an "A" check every 40-45 hours of flying, and not more than three to four calendar days apart. Aircraft maintenance can be performed only at a few stations which depend on the fleet type. This is due to the capital intensive nature of the maintenance equipment, as well as the large costs incurred in training people for the high-skills job of aircraft maintenance [Wells 1994].

Some major carriers have an additional requirement that each aircraft in the fleet

[2]Source: World Wide Web URL: http://www.amrcorp.com/aa_home/servinfo/maint_pro.htm

fly each flight leg assigned to that fleet. In other words, the rotation flown by each aircraft should be an Euler tour. An Euler tour is a path that starts and ends at the same node in a graph, while covering all the arcs in the graph exactly once [Larson and Odoni 1981]. If each flight in the schedule was represented by an arc, and the starting and ending city represented by the start and end nodes of that arc, then an Euler tour would be a rotation that covers all the flights [Clarke et al 1995] though the rotations may not be necessarily feasible with respect to maintenance regulations. Some airlines have the Euler tour requirement for operational reasons, such as equal utilization of all aircraft in a fleet [Barnhart 1996].

4.4.3 Earlier Work

Because most earlier work on "A" check maintenance routing has enforced the Euler tour requirement, most of the work has been heuristic [Feo and Bard 1989, Kabbani and Patty, 1992, Talluri 1994, Talluri 1994b, Talluri and Gopalan 1995, Gopalan and Talluri 1995, Clarke et al 1995, Boland and Clarke 1996]. Clarke et al [1995] present a solution approach in which routing is done with through flight assignment and different kinds of maintenance stops.

Our aircraft maintenance routing model will be shown to be a special case of the *aircraft scheduling model*, discussed in the next section.

4.5 The Aircraft Scheduling Model

We refer to the combined fleet assignment, through assignment, and aircraft maintenance routing model, as simply the *aircraft scheduling model*. The networks at the core of our model are time-line networks and connection networks (section 2.3.1).

4.5.1 The Aircraft Scheduling Formulation

Since the objective of the aircraft scheduling problem is to assign simultaneously fleets to flights and rotations to aircraft, variables representing individual flights are

no longer sufficient to capture all the interdependencies between flights. So we define new variables called *strings*, i.e., sequences of flights that start and end at maintenance stations.

String Properties

A string is a partial rotation and has the following properties:

- The origin of the first flight and the destination of the last flight in the sequence are maintenance stations for that fleet,
- the destination of a flight is the origin of the next flight in the sequence,
- the flying time and elapsed time of the sequence does not exceed the maximum time-between-maintenance limits set by the FAA and by the airline,
- the sequence satisfies any additional constraints imposed by the airline, such as maximum flying time per day, etc., and
- the starting time of the string is the starting time of the first flight in the sequence; the ending time of the string is the *maintenance ready time* of the last flight in the sequence for that fleet, where the maintenance ready time of a flight is defined as the time that the aircraft flying that flight will be ready after it has been serviced with a "A" check.

Example 10 *Figure 4-5 shows a string* **S1**, *corresponding to the sequence* $\{\mathbf{F1}, \mathbf{F2}\}$. *Other examples of strings in the same network are* **S2** *and* **S3**, *from the sequences* $\{\mathbf{F3}, \mathbf{F4}\}$ *and* $\{\mathbf{F1}, \mathbf{F2}, \mathbf{F3}, \mathbf{F4}\}$ *respectively.*

One variable is assigned to each string in each fleet. As before, constraints are placed on the string variables that enforce *coverage*, i.e., each flight should be assigned to exactly one string of only one fleet, *balance*, i.e., each string of a fleet type ending at an airport must be balanced by a string of the same fleet type leaving that airport, and *aircraft count*, i.e., the number of aircraft of each fleet used by the solution does

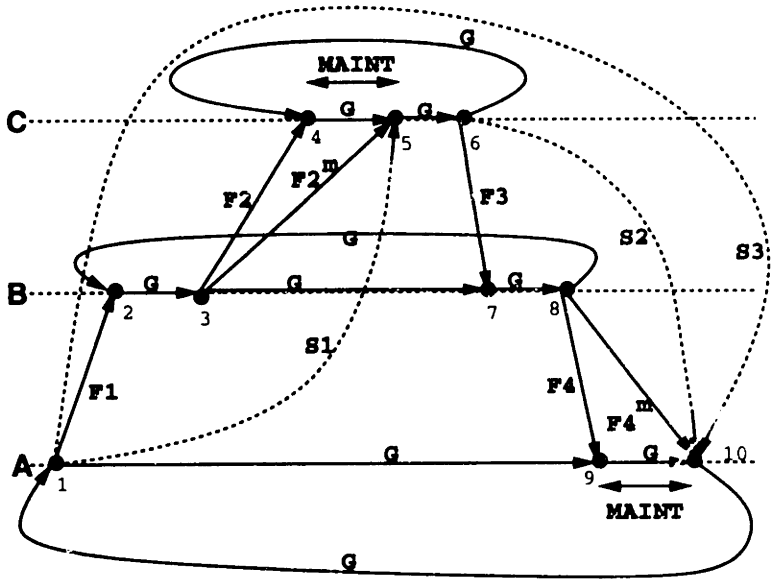

Figure 4-5: String Generation

not exceed the number available in each fleet. Constraints for maintenance are not needed since each string has a period of maintenance affixed to its end.

The String Based Model

The aircraft scheduling model can be represented by the mixed integer program shown in Equations (4.15)-(4.20). The notation for this model is given in Table 4.3.

$$\min \sum_{k \in K} \sum_{j \in J_k} c_{jk} x_{jk} \tag{4.15}$$

$$s.t.$$

$$\sum_{k \in K} \sum_{j \in J_k} a_{ijk} x_{jk} = 1 \quad \forall i \in F \tag{4.16}$$

$$\sum_{j \in J_k} b1_{ljk} x_{jk} + \sum_{g \in G_k} b2_{lgk} y_{gk} = 0 \quad \forall l \in L_k, \forall k \in K \tag{4.17}$$

$$\sum_{j \in J_k} d1_{jk} x_{jk} + \sum_{g \in G_k} d2_{gk} y_{gk} \leq N_k \quad \forall k \in K \tag{4.18}$$

$$x_{jk} \in \{0,1\} \forall j \in J_k, \forall k \in K \tag{4.19}$$

$$y_{gk} \geq 0 \quad \forall g \in G_k, \forall k \in K \tag{4.20}$$

F	=	Set of Flights
K	=	Set of Fleets
N_k	=	Number of aircraft in fleet
J_k	=	Set of all strings in fleet k
G_k	=	Set of all ground arcs in fleet k
L_k	=	Set of maintenance stops for fleet k
e_{ik}	=	Operating cost of flight i flown by fleet k
S_{jk}	=	Set of flights in string j of fleet k
c_{jk}	=	cost of string j in fleet k
e_{ik}	=	cost of flight i in fleet k
x_{jk}	=	1, if string j in fleet k is used, 0 otherwise
a_{ijk}	=	1, if flight i covered by string j in fleet k, 0 otherwise
y_{gk}	=	Number of aircraft on ground arc g in fleet k
$b1_{ljk}$	=	1, if string x_{jk} begins at node l,
	=	-1, if string x_{jk} ends at node l, 0 otherwise
$b2_{ljk}$	=	1, if ground arc y_{gk} begins at node l,
	=	-1, if ground arc y_{gk} ends at node l, 0 otherwise
$d1_{jk}$	=	1, if string x_{jk} crosses the count time, 0 otherwise
$d2_{gk}$	=	1, if ground arc y_{gk} crosses the count time, 0 otherwise

Table 4.3: Notation for the Aircraft Scheduling Model

The objective function (4.15) minimizes the total cost of all strings used, where as described earlier, optimization involves finding the best balance between operating costs, spill costs, and through revenues. Equations (4.16) ensure that each flight is assigned to exactly one string. This implicitly assigns a flight to a specific fleet type. Equations (4.17), one set for each fleet type, ensure that continuity is maintained between the strings and the ground arcs in each fleet. Equations (4.18), one for each fleet type, ensure that the number of aircraft used by a fleet does not exceed the number of aircraft available for that fleet. Equations (4.19) and (4.20) maintain integrality of strings and ground arcs. The integrality of the strings will ensure that the ground arcs have integral flow.

A similar model has been developed for flight scheduling with time-windows [Desaulniers et al 1994b]. In their model, they use variables, called *schedules*, representing sequences of flights over a one day period. However, their goal is primarily to decide the start time of flights within time-windows while simultaneously assigning fleets to flights. Maintenance considerations cannot be modeled explicitly with this approach.

Strings Vs. Rotations

Since strings are partial rotations, the optimal set of strings have to be connected together to generate complete aircraft rotations. Instead of a model with string variables, an alternative might be to develop a model using variables that represent entire, rather than partial, rotations. We opted to model the problem using partial rotations as our decision variables for the following reasons:

1. A partial rotation, or string, is limited to be at most 3-4 days long. The maximum length of a rotation in a weekly problem, however, is equal to N_k weeks, where N_k is the number of aircraft in a given fleet. This would complicate the solution process significantly.

2. If rotations are used as variables, the balance equations (4.17) in the aircraft scheduling model [Equations (4.15) - (4.20)] can be eliminated. While the number of constraints is reduced somewhat, the number of variables grows enormously. Secondly, since each rotation is comprised of possibly several strings, many more columns are necessary in the rotation formulation than in the string formulation to represent the same number of possible solutions. This implies that many more variables will have to be considered in solving the rotation formulation than the string formulation. The expected result is that the rotation formulation would take a prohibitive amount of time to solve. This follows even if rotation columns can be generated efficiently, and for the reasons stated above, this is not likely.

In both the aircraft and crew scheduling problems, variables are defined to simplify constraints that are difficult to formulate, like maintenance constraints in strings, or rest requirements in a duty. At the same time, the definition of variables should be such that the growth of the number of variables is limited to the greatest extent possible, i.e., variables representing strings are preferred over those representing rotations.

4.5.2 Model Properties

We can generate rotations from the strings in any solution to the aircraft scheduling model using a simple flow decomposition method [Ahuja et al 1993]. By assigning the aircraft of a particular fleet to the rotations of that fleet, we ensure that the aircraft are maintained as required. Although some airlines require an Euler tour, this requirement is not modeled in our aircraft scheduling model. If an Euler tour is required, additional constraints may have to be added to our aircraft scheduling model such as those in Boland and Clarke [1996].

Through revenues can be modeled easily in our aircraft scheduling formulation by adding, to the cost of a string, the (negative of the) through revenues accrued by successive flight pairs in a string. Then, any sequence of flights in the optimal set of strings can be assigned to a through. Our approach has limitations in that interdependencies between throughs cannot be captured. For example, there may be more than one possible itinerary that serves an origin-destination market during a specific time window on a specific day. But through revenues are accrued for only one itinerary. Such restrictions can be modeled by adding constraints that limit the model to choose at most one of competing itineraries [Talluri and Gopalan 1995]. In a long-haul schedule however, the number of competing through assignments for a given origin-destination market is small, and so the number of constraints to be added will also be small. We do not add any of these constraints in our model.

4.5.3 Maintenance Routing - Special Case of the Aircraft Scheduling Problem

Consider the aircraft scheduling problem [Equations (4.15) - (4.20)] with only one fleet type. This is a situation that one would encounter if the fleet assignment decision has been made and the flights partitioned into separate fleets. The aircraft in each fleet have to be routed so that each of them are maintained regularly and through revenues are maximized. This model, which we refer to as the *through-maintenance-routing* model, is given in Equations (4.21)-(4.26), and the notation is given in Table 4.4.

F	=	Set of Flights
N	=	Number of aircraft
J	=	Set of all strings
G	=	Set of all ground arcs
L	=	Set of maintenance stops
S_j	=	Set of flights in string j
c_j	=	cost of string j
x_j	=	1, if string j is used, 0 otherwise
a_{ij}	=	1, if flight i covered by string j, 0 otherwise
y_g	=	Number of aircraft on ground arc g
$b1_{lj}$	=	1, if string x_j begins at node l,
	=	-1, if string x_j ends at node l, 0 otherwise
$b2_{lj}$	=	1, if ground arc y_g begins at node l,
	=	-1, if ground arc y_g ends at node l, 0 otherwise
$d1_j$	=	1, if string x_j crosses the count time, 0 otherwise
$d2_g$	=	1, if ground arc y_g crosses the count time, 0 otherwise

Table 4.4: Notation for the Through-Maintenance-Routing Model

The underlying network for this model is the same as that for the aircraft scheduling model (section 4.2.3).

$$\min \sum_{j \in J} c_j x_j \tag{4.21}$$

s.t.

$$\sum_{j \in J} a_{ij} x_j = 1 \quad \forall i \in F \tag{4.22}$$

$$\sum_{j \in J} b1_{lj} x_j + \sum_{g \in G} b2_{lg} y_g = 0 \quad \forall l \in L \tag{4.23}$$

$$\sum_{j \in J} d1_j x_j + \sum_{g \in G} d2_g y_g \leq N \tag{4.24}$$

$$x_j \in \{0, 1\} \; \forall j \in J \tag{4.25}$$

$$y_g \geq 0 \quad \forall g \in G \tag{4.26}$$

This model contains strings of only one fleet type (hence, the fleet subscript k has been dropped). The objective function (4.21) minimizes the total cost of the strings used. Note that the cost of flying an aircraft is independent of the string which is

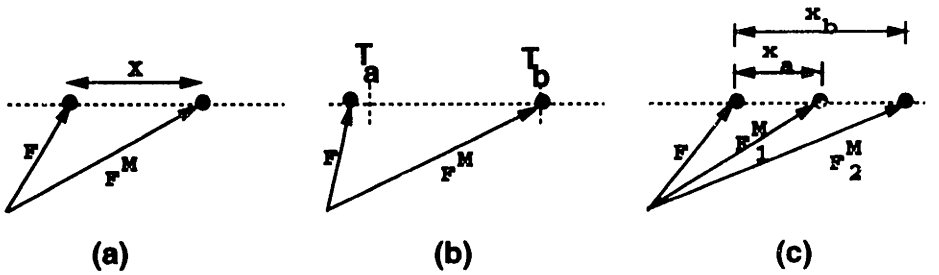

Figure 4-6: Examples of Maintenance Modeling

assigned to it since all aircraft in this through-maintenance-routing problem are flown by the same fleet type. So the cost of flying a flight is taken out of this model, and the cost c_j of string j is simply the (negative of) the through revenues that can be achieved by flying the string.

Equations (4.22) ensure that all flights assigned to a specific fleet are flown by exactly one aircraft. Equations (4.23) ensure aircraft continuity and Equations (4.24) ensure that the number of aircraft used by the strings do not exceed the number available. Equations (4.25) and (4.26) ensure the integrality of the strings and the ground arcs even though Equation (4.26) is not an integrality constraint. The integrality of the ground arcs is automatically satisfied by a balanced schedule, i.e., one with the same number of arrivals and departures. We include ground arcs and aircraft continuity constraints (4.23) to determine the number of aircraft on the ground at any point in time so that the aircraft count constraints (4.24) can be modeled.

4.5.4 Model Versatility

A variety of restrictions can be incorporated into the aircraft scheduling model:

1. **Landing Restrictions:** If a particular fleet type k is not allowed to fly a flight i, for reasons such as gate limitations, noise restrictions, etc., then we let the operating cost of the flight for that fleet type be very high. This will prevent any strings in fleet k from flying flight i.

2. **Turn Times:** The structure of the flight network allows the turn time to be adjusted for each fleet type and airport, since the ending time of a flight arc in the underlying networks equals the sum of the flight's ending time and the turn time for that flight at that station and aircraft type (section 2.3.1).

3. **Maintenance Criterion:** Since maintenance is added to the end of the string using the time-line flight network (section 4.2.3), various maintenance scenarios can be modeled by modifying the times of the ending nodes of the maintenance arcs.

   (a) *x hours maintenance starting at an arrival:* The time of the ending node of the maintenance arc is set equal to x hours after the arrival of the corresponding flight. In Fig. 4-6(a), the ending node for maintenance arc F^M is x hours after the end of flight F. Note that x can be modified for each flight, fleet type, or airport as the case may be.

   (b) *Maintenance within time-windows:* If the flight ends at a maintenance station after the start of the time-window, maintenance is typically not allowed. If it ends before the start of the time-window, then we set the time of the ending node of the maintenance arc to equal the end of the time window. In Fig. 4-6(b), since flight F ends before T_a, the ending node of arc F^M is created at T_b.

   (c) *Multiple maintenance criterion:* Consider an example in which the maintenance of an aircraft depends on the total flying hours since the last "A" check. In such a case, one maintenance arc is created for each scenario, giving adequate time for the maintenance corresponding to that scenario.

   Example 11 *Consider a case where there are two possible maintenance checks that can be performed at a station, one that is x_a hours long, and another that is x_b hours long depending on the flying time of the aircraft since the last check. Figure 4-6(c) shows that two maintenance arcs are created for flight* $\mathbf{F}$, $\mathbf{F_1^M}$ *and* $\mathbf{F_2^M}$, *which end x_a and x_b hours respectively after the end of flight* $\mathbf{F}$.

4.6 Solving the Aircraft Scheduling Model

The aircraft scheduling model [Equations (4.15)-(4.20)] is solved using *branch-and-price* (section 2.4) and *column generation* (section 2.2).

4.6.1 LP Solution to the Aircraft Scheduling Model

Since it is impractical to explicitly enumerate all feasible strings, column generation is necessary to solve the LP relaxation of the aircraft scheduling problem. The column generation algorithm for the aircraft scheduling problem is:

- *Step 0 - Create Initial Feasible Solution:* An initial feasible solution is created using a set of high-cost artificial variables, one for each flight. This, alongwith the ground arc variables, is the initial *restricted master problem.*
- *Step 1 - Solve the Restricted Master Problem:* Find the optimal solution to the current restricted master problem containing only a subset of all strings. The optimal dual variables are used to generate more variables (strings).
- *Step 2 - Solve the Pricing Problem:* Generate strings with negative reduced cost. If no strings are generated, STOP: the LP is solved.
- *Step 3 - Construct a New Restricted Master Problem*– Add the strings generated by the pricing problem to the restricted master problem; go to Step 1.

While steps 1 and 3 can be carried out using optimization software such as CPLEX [Bixby 1994] or OSL [Druckerman et al 1991], step 2 (the pricing problem) should be tailored to exploit problem structure. Pricing problems have been discussed in section 2.3.

4.6.2 The Pricing Problem Solution (Step 2)

The pricing problem finds one string for each node pair for each fleet type. The reduced cost of a string is

$$\bar{c_{jk}} = \overbrace{c_{jk} - \sum_{i \in S_{jk}} \pi_i}^{\hat{c_{jk}}} + \overbrace{\lambda^s_{jk} - \lambda^e_{jk} - d1_{jk}\sigma_k}^{\mu_{sek}} \tag{4.27}$$

where λ^s_{jk} and λ^e_{jk} are the dual variables for the rows corresponding to balance constraints for nodes s and e, and σ_k is the dual for the count constraint for fleet k. While μ_{sek} is constant for a node pair $\{s, e\}$, $\hat{c_{jk}}$ varies for each flight sequence. To find negative reduced cost strings, we carry out the following steps for *each node pair* in the flight network.

1. For the node pair, find the string with the minimum value of $\hat{c_{jk}}$.

2. For that node pair, compute μ_{sek}.

3. If $\bar{c_{jk}} = \hat{c_{jk}} + \mu_{sek} < 0$, we have a string with negative reduced cost.

If the reduced cost of a string is negative, a column corresponding to that string is added to the restricted master problem. See section 2.3 for details about pricing problems.

4.6.3 Type of Pricing Problem - Maintenance Criterion

The type of shortest path procedure will vary depending on the specific maintenance requirements enforced. If the *maximum elapsed time* requirement is enforced, i.e, maintenance must occur for each aircraft at least once in every three to four days, then the pricing problem can be modeled as a simple (unconstrained) shortest path problem. The elapsed time requirement is modeled by eliminating from consideration those node pairs whose elapsed time exceeds this limit. Then, the minimum reduced cost paths satisfying the maximum elapsed time rule are the set of shortest paths between the *allowable* node pairs. These paths are easy to identify using label-setting

or label-correcting shortest path procedures, like those described in Deo and Pang [1984] and Ahuja et al [1993].

If the *maximum flying time* requirement is enforced, then maintenance must occur for each aircraft at least once in every 40 to 45 flying hours and the pricing problem can be modeled as a constrained shortest path problem. Constrained shortest path problems are those where the shortest path between two nodes is required to satisfy certain additional constraints[Desrochers and Soumis 1988, 1988b]. In this case, the objective is to find the shortest path that contains at most 45 hours of flying. Examples of simple and constrained shortest path problems are given in sections 2.3.3 and 2.3.4.

4.6.4 IP Solution to the Aircraft Scheduling Model

If the string variables in the aircraft scheduling model solution are integral, then the ground arc variables will automatically be integral. This implies that we need only ensure the integrality of the strings. We refer to the branching rules we use as the *fleet-flight pair* rule and *follow-on* rule. The first rule is a special case of a rule mentioned in Barnhart el al [1995b]. The second rule was suggested by Ryan and Foster [1981], and its validity is also provided in Barnhart et al [1995b]. The use of these two rules will eliminate any fractional solutions. We now describe these rules and prove that they guarantee that optimal integer solutions will be found.

Fleet-Flight Pair Rule

This branching rule assigns each flight to exactly one fleet type. The procedure for accomplishing this is described by the following steps:

1. *Step 1:* If the solution is not fractional, we have an integral solution. If it is fractional, calculate

$$f_{ik} = \sum_{j \in J_k} a_{ijk} x_{jk} \quad \forall i \in N,\ \forall k \in K \tag{4.28}$$

where f_{ik} can be defined as the total *flow* of fleet k on flight i. If $f_{ik} = 1$, then fleet k is assigned to flight i. Note that $0 \leq f_{ik} \leq 1 \; \forall i \in N, \forall k \in K$.

2. *Step 2:* If $f_{ik} \in \{0, 1\} \; \forall i \in N, \forall k \in K$, then STOP - we have partitioned the problem into K fleets, and each of these resulting *through-maintenance-routing* problems can be solved using the *follow-on* branching rule discussed in the next section. Otherwise, go to *Step 3.*

3. *Step 3:* Identify a flight i^* and fleet k^* for which $0 < f_{i^*k^*} < 1$, and create two branches, as follows:

   - *Right Branch:* On this branch, set

   $$f_{i^*k^*} = \sum_{j \in J_k} a_{i^*jk^*} x_{jk^*} = 1 \tag{4.29}$$

     This rule requires that flight i^* be assigned to fleet k^*. The constraint matrix is modified by setting the lower and upper bounds to zero for all those strings that fly flight i^* but do not belong to fleet k^*. The pricing problems are modified by setting $e_{i^*k} = \infty \; \forall k \in K, k \neq k^*$, ensuring that only strings of fleet k^* will contain flight i^*.

   - *Left Branch:* On this branch, set

   $$f_{i^*k^*} = 0. \tag{4.30}$$

     This rule requires that flight i^* not be assigned to fleet k^*. The constraint matrix is modified by setting the lower and upper bounds to zero for all the strings of fleet k^* that fly flight i^*. The pricing problems are modified by setting $e_{i^*k^*} = \infty$ ensuring that no string in fleet k^* will fly flight i^*.

   After the new LP is solved by column generation, return to *Step 1.*

It is easy to see that this branching rule will result in partition of flights into fleets.

Follow-On Rule

The *fleet-flight pair* rule terminates by either generating an integer solution, in which case we have solved the aircraft scheduling problem, or we have partitioned the problem into several through-maintenance-routing problems, one for each fleet. The through-maintenance-routing model is a set partitioning model with extra balance and count constraints [(4.23)-(4.24)]. Barnhart et al [1995b] show that integer solutions can be achieved for set partitioning problems by using a follow-on branching rule. For this application, the follow-on rule ensures that each flight in an aircraft's rotation is followed by exactly one flight. The steps to achieve optimal integer solutions to the through-maintenance-routing problems using follow-on branching rules are (we drop the fleet subscript k since each problem is for a single fleet):

1. **Step 1:** If the solution is not fractional, the current through-maintenance-routing problem is solved. If the solution is fractional, identify a fractional string j^* such that $0 < x_{j^*} < 1$.

2. **Step 2:** Consider a flight i_1 for which $a_{i_1 j^*} = 1$. There must be another string j^{**} such that $a_{i_1 j^{**}} = 1$ and $0 < x_{j^{**}} < 1$.

3. **Step 3:** Compare the strings j^* and j^{**} and identify flights i_2 and i_3 so that i_3 follows i_2, i_2 appears in both strings j^* and j^{**}, and i_3 appears in only one of those strings. It is always possible to identify such a pair because no two strings in the same fleet can have the same sequences of flights. Define $FO_r \subset J$ as the set of all strings which either contain both flights i_2 and i_3 or contain neither, and let $FO_l \subset J$ be the set of strings that do not contain both flights i_2 and i_3. Now create two branches as follows:

   - *Right Branch:* On this branch, set

$$\sum_{j \in FO_r} x_j = 1 \tag{4.31}$$

In this branch, flights i_2 and i_3 will either appear together in a string, or will both be absent from a string. The pricing problem is modified by simply eliminating all connections from i_2 that do not lead to i_3, and eliminating all connections to i_3 that do not begin at i_2. In addition, we should ensure that no string begins with flight i_3 or ends with flight i_2. Then, forbidden strings will not be generated by the pricing problem solver. In addition, the constraint matrix is modified by setting the lower and upper bounds to zero of all the strings that fly i_2 or i_3 but not both.

- *Left Branch:* On this branch, set

$$\sum_{j \in FO_l} x_j = 1 \tag{4.32}$$

In this branch, flights i_2 and i_3 will not appear together in a string. The pricing problem can be modified by simply eliminating the connection between flights i_2 and i_3. This will ensure that no string will fly both i_2 and i_3. In addition, the constraint matrix is modified by setting the lower and upper bounds to zero of all strings that fly both flights i_2 and i_3.

In summary, branching on the fleet-flight pair rule ensures that each flight is assigned to exactly one fleet. This decomposes the problem into K *through maintenance routing* problems, one for each of the K fleets. The follow-on branching rule then ensures that each flight in a rotation is followed by exactly one flight, and applying the proof of Barnhart et al [1995b], this branching rule will generate optimal integer solutions to the individual through-maintenance-routing problems for each fleet. The end result is that the original aircraft scheduling problem is solved using these two branching strategies.

Description	Value
Long-Haul Flights	858
Domestic Flights	266
Total Flights	1,124
Total Cities	40
Number of Fleets	9
Total Aircraft Count	89

Table 4.5: Data Characteristics

4.7 Proof of Concept

All computations have been run on an IBM RS-6000/370 using the CPLEX v3.0 callable library[Bixby 1994], and MINTO v.2.0a [Savelsbergh and Nemhauser, 1996].

4.7.1 Data Characteristics

Our computations were based on a dataset provided by a long-haul carrier. Table 4.5 shows that the flights comprise a weekly schedule and include 40 cities. There is a total of 89 aircraft belonging to 9 fleet types. While the total number of flights is 1124, only 858 of them are international flights, i.e., either start or end at a non-U.S. airport. The remaining 266 domestic flights are added so that the aircraft flying the long-haul flights can reach their domestic maintenance stations. In our computations, we assumed that the domestic flights need be covered only if necessary to find feasible routings to cover the international flights. This was modeled by adding zero cost artificial variables for the domestic flights.

The objective function value for each fleet-flight pair requires the use of proprietary and sensitive airline data. As a result, proxy costs are used. The cost of each flight flown by a specific fleet is set equal to the product of the total seat miles flown by the flight using that fleet and a cost per seat mile.

Description	Value
Solution Times	
Time to LP	42m 44s
Time to IP	2h 10m 29s
Total Time	2h 53m 13s

Table 4.6: Fleet Assignment Model Results

4.7.2 Results of Sequential Solution

Table 4.6 shows that the linear programming (LP) relaxation[3] to the enhanced fleet assignment model (section 4.2.4) takes about 43 minutes to solve. The Integer Programming (IP) solution to this problem was difficult to obtain, partly because of the size of the model. The model had 7,155 constraints and 18,777 variables, of which 13,188 were integer variables. Our initial approach[4] did not yield any integer solution after about 12 hours of computation time and searching 500 nodes.

Our next approach was to first solve the LP relaxation, and then fix flight variables to unity if their value in the LP solution was close to unity. 572 flights were fixed at unity using this method. The resulting reduced size IP was then solved and a feasible solution was obtained in a little over 2 hours. The solution was stopped with the first IP solution and the total time was just under three hours. This approach generates a feasible solution with a reasonable gap of 1.6%, where the gap is defined as the difference in the IP and LP solution values divided by the LP solution value.

Using the data in Table 4.5, and the fleeting decisions from above, the problem was decomposed into 9 through-maintenance-routing subproblems, one for each fleet. The through-maintenance-routing model [Equations (4.21) - (4.26)] without through values was used to try to obtain maintenance feasible solutions in each of the 9 problems. Feasible routings could not be obtained in five of these problems. This shows that the sequential solution of fleet assignment and through-maintenance-routing will

[3]The fleet assignment LP was solved using CPLEX's dual simplex with the steepest edge option. In addition, the following options were used - *setadvind(1)*, *setpreind(1)*, and *setaggind(1)*.

[4]The options used to solve the IP are *setcoeredind(1)*, *setcliques(1)*, *setcovers(1)*, *setheuristic(1)*, and *setndlim(500)*.

Description	Value
Formulation	
Flight Cover Constraints	1,124
Total Balance Constraints	648
Fleet Count Constraints	9
Total Constraints	1,781

Table 4.7: Aircraft Scheduling Formulation

not always yield feasible maintenance rotations for long-haul operations.

4.7.3 Results for the Aircraft Scheduling Model

Table 4.7 shows that the aircraft scheduling model formulation has 1781 constraints of which the number of balance constraints is only 648. Without node aggregation (section 2.3.1), the number of balance constraints is 5,424 and the total number of constraints is 6557. Hence, node consolidation reduces the number of constraints by about 73%. An interesting observation is that the number of ground arcs in the formulation is equal to the number of balance constraints, because there is one node for each balance constraint, and there is one ground arc (or wrap around arc) for each node. The implication of this is that the number of ground arcs required in the formulation is also decreased from 5,424 to 648.

LP Solution

The LP relaxation to the root node of the aircraft scheduling model was solved using CPLEX. The initial starting solution was a set of artificial variables, one for each flight. Since the domestic flights are added for feasibility but do not need to be covered, they are included in the flight cover constraints, but the artificial variable corresponding to these flights are given zero costs. This ensures that they will be used only if needed, and will not be used more than once.

The problem size is managed by deleting columns regularly. Each time the number of columns exceeds a predetermined limit (in our case, 10,000 columns), all variables

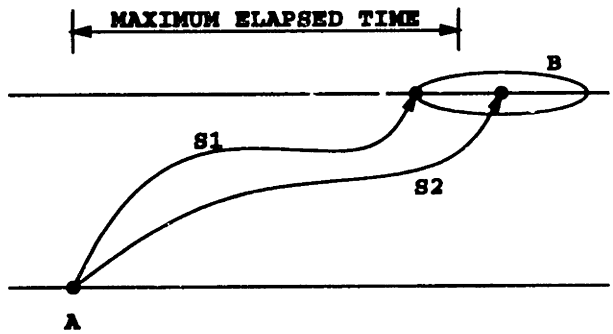

Figure 4-7: Drawback of Time Line Networks

are deleted with the exception of the ground arc variables, the artificial variables, and the basic variables.

The column generation algorithm uses both the time line flight network as well as the connection network (section 2.3.1). Although the time line network is compact after node consolidation, it has the following drawback.

Example 12 *Figure 4-7 shows a time line of two cities. There are two strings* **S1** *and* **S2** *going from node* **A** *to node* **B** *(which is an aggregated node). String* **S1** *satisfies the maximum time limit between maintenance stops while* **S2** *does not. If the shortest path between node pair* {**A**,**B**} *is* **S2**, *then no string with a negative reduced cost will be identified because* **S2** *violates maintenance requirements. This may be a problem because the feasible string* **S1** *may also have a negative reduced cost.*

Since each flight arc has unique start and end nodes in the connection network, a shortest path algorithm between each node pair eliminates this problem. Unfortunately, the connection network can quickly grow in size as the number of connections increases dramatically with the number of flights. This number however, can be reduced using islands (section 4.2.3). Care should be taken not to eliminate connections that have through values assigned to them.

In all of our computational work, the elapsed time between maintenance stops is limited to 4 days, and no limits are placed on the flying time. Table 4.8 shows the results of using three different approaches to solving the LP.

- **Case 1 - Base Case:** All iterations of column generation are solved using the

Description	Case 1	Case 2	Case 3
Iterations	191	160	204
Total Columns	191,687	168,566	88,049
Column Deletions	23	20	11
Time	8h 33m 26s	5h 38m 4s	4h 37m 9s

Table 4.8: Aircraft Scheduling LP

connection network. Islands are not used, and the number of arcs (connections) is 69,290. This method takes 191 iterations to reach optimality during which time it generates 191,687 columns, and the column deletion is done 23 times. The LP takes about eight and a half hours to solve.

- **Case 2 - Base Case with Islands:** Using islands, the number of connections are reduced to 44,340 - a reduction of 36%. It takes fewer iterations (160), generates fewer columns (168,566), deletes columns fewer times (20), and takes under 6 hours to solve - a reduction in time of 34% from the base case. The objective function value achieved in this case is the same as the base case.

- **Case 3 - Hybrid Approach:** Initially columns are generated using the time line network and columns that violate maintenance requirements are deleted. When no additional columns are generated with the time line network, we switch to generating columns as in case 2, i.e., using the connection network with islands. Although this hybrid approach takes 204 iterations, the LP spends the first 173 iterations and 85% of the time in the time line network. As a result, fewer of the more expensive iterations using the connection network are required, and this hybrid approach generates fewer columns (88,049), deletes columns fewer times (11), and takes about four and half hours to complete - a reduction of 46% over the base case.

Description	Tol.= 1.00%	Tol.= 0.75%	Tol.= 0.50%	Tol.= 0.25%	Tol.= 0.05%
Branch-and-Price					
Follow-on Nodes	176	168	200	107	282
Fleet-Flight Nodes	34	46	15	25	21
Total B&P Nodes	210	214	215	132	303
The Solution					
Final Columns	11054	11303	12395	13882	15865
Cost of IP(B)	102.36%	102.50%	101.91%	101.52%	101.65%
Gap (B)-(A)	2.36%	2.50%	1.91%	1.52%	1.65%
Comp. Times					
Time to Solve LP only	4h 37m	4h 37m	4h 37m	4h 37m	4h 37m
Time to Solve IP only	51m	1h 1m	1h 14m	1h 2m	3h 38m
Total Time	5h 28m	5h 38m	5h 51m	5h 39m	8h 15m

Table 4.9: Aircraft Scheduling IP

IP Solution

MINTO was used to implement our branch-and-price approach to solve the IP, but not the LP because MINTO, unlike CPLEX, does not allow column deletion. To get an IP solution, we used the branching strategy of first *fixing follow-ons* and then *fixing fleet-flight pairs.*

Branch-and-price columns are generated at potentially every node of the search tree. Obviously, this can increase the solution time significantly. Instead, we define a certain tolerance and generate columns at a node only when the solution value at that node exceeds the root node LP solution value by that tolerance.

Table 4.9 shows the results obtained under five different tolerance values. In all cases, the IP is started from the final LP matrix from case 3 of the previous section, and the IP solution is terminated when the first IP solution is obtained. The optimality gaps[5] vary from 1.52% to 2.50%. While the times taken to solve the IP do not vary significantly for tolerance values above 0.25%, the IP takes much longer

[5]To avoid divulging the solution values, all solution values are divided by the optimal LP solution value of the aircraft scheduling model and expressed as a percentage. Thus "(A)", i.e., the optimal LP solution value is 100.00%.

when the tolerance is reduced to 0.05%. Not only does it take much longer, it also does not show an improvement in solution value. The number of domestic flights used by these solutions vary between 40 and 45, not a great variation. An interesting result is that when the tolerance is reduced, the number of columns in the final IP solution increases.

In all the cases in Table 4.9, a limit of 20 was placed on the number of non-integer follow-ons that could be fixed per branch-and-price node, and a similar limit of 5 was placed on the number of non-integer fleet-flight assignments. Our experience indicated that the solution was more sensitive to the number of fleet-flight pairs fixed at each node. That is why the limit was kept low at 5. However, judging by the number of branch-and-price nodes using the fleet-flight pair rule, this does not seem to be a problem.

4.7.4 Comparisons of Fleeting and Aircraft Scheduling

Table 4.10 compares the results for the fleet assignment model [Equations (4.7) - (4.14)] and the aircraft scheduling model [Equations (4.15) - (4.20)]. Neither model uses through values. The difference between the LP's of the fleet assignment model and the aircraft scheduling model is 0.23%, and the gap between the IP's is 0.15%. This is interesting considering that the fleet assignment LP and IP can be obtained significantly faster, i.e., more than six (two) times faster than the corresponding LP (IP) of the aircraft schedulng model.

While the fleet assignment model cannot be used to generate feasible rotations easily, it can be used as a *surrogate* model, in lieu of the aircraft scheduling model, if a quick estimate is needed of the fleet assignment cost of a proposed schedule. To show the accuracy of the fleet assignment model lower bound, we consider two other methods of obtaining lower bounds. In the first, each flight is assumed to be assigned to the least cost fleet without any considerations of fleet count or flow balance. The second method uses the LP solution to the fleet assignment model without the additional constraints requiring an adequate number of maintenance opportunities [Equations (4.1) - (4.6)]. The values of the lower bounds for the two

Description	Aircraft Scheduling	Fleet Assignment
LP Value	100.0%	99.77%
Total Time to LP	4h 37m 9s	42m 44s
IP Value	101.52%	101.37%
Total Time to IP	5h 39m 23s	2h 53m 13s

Table 4.10: Comparisons of Fleeting and Aircraft Scheduling

$$
\begin{aligned}
ST_k &= \text{List of maintenance stations for fleet } k \\
u_{sk} &= 1 \text{ if station } s \text{ can service fleet } k\text{, 0 otherwise} \\
U_{sk} &= \text{fixed cost of providing maintenance for fleet } k \text{ at city } s \\
v_{sk} &= \text{Number of maintenance checks for fleet } k \text{ at station } s \\
V_{sk} &= \text{variable cost of one maintenance check at city } s \text{ for fleet } k \\
MAX_{sk} &= \text{Maximum Number of fleet } k \text{ planes that can be serviced at city } s \\
MIN_{sk} &= \text{Minimum Number of fleet } k \text{ planes that can be serviced at city } s \\
END_{sjk} &= 1 \text{ if string } j \text{ in fleet } k \text{ ends at station } s\text{, 0 otherwise}
\end{aligned}
$$

Table 4.11: Notation for Maintenance Base Planning

methods are 61.19% and 94.20% of the aircraft scheduling LP optimal solution value respectively[6]. This shows that our fleet assignment model [Equations (4.7) - (4.14)] is highly accurate at estimating fleet assignment costs for a given flight schedule.

4.8 Maintenance Base Planning

We define the maintenance base planning problem as that of selecting airports to serve as maintenance stations for each fleet, and assigning maintenance visits for each fleet to each station [Feo and Bard 1989].

4.8.1 Maintenance Base Planning Inputs

In addition to the inputs to the aircraft scheduling problem, the maintenance base planning model has a set of *fixed* and *variable* costs as inputs [Feo and Bard 1989]. The

[6]While the first bound can be trivially calculated from the data, the second bound took under 17 minutes to compute.

variable costs are those incurred for each maintenance visit, and may include wages, costs of parts replaced, or any other costs incurred only during that specific visit. The fixed costs refer to one-time costs incurred in allowing a fleet to be maintained at a given maintenance base, and may include inventory costs, costs of relocating maintenance staff, costs of building new hangars, etc. In addition, there is an upper and lower limit on the total number of aircraft of a particular fleet that can be maintained at a given station. The upper limit could be due to hangar capacity, and the lower limits could be labor contract requirements, etc.

4.8.2 The Model

The aircraft scheduling model can be extended to create the maintenance base planning model given in Equations (4.33) - (4.43). The notation for this model is given in Table 4.11.

$$\min \sum_{k\in K}\sum_{j\in J_k} c_{jk}x_{jk} + \sum_{k\in K}\sum_{s\in S_k}(U_{sk}u_{sk} + V_{sk}v_{sk}) \tag{4.33}$$

$s.t.$

$$\sum_{k\in K}\sum_{j\in J_k} a_{ijk}x_{jk} = 1 \quad \forall i \in F \tag{4.34}$$

$$\sum_{j\in J_k} b1_{ljk}x_{jk} + \sum_{g\in G_k} b2_{lgk}y_{gk} = 0 \quad \forall l \in L_k, \forall k \in K \tag{4.35}$$

$$\sum_{j\in J_k} d1_{jk}x_{jk} + \sum_{g\in G_k} d2_{gk}y_{gk} \leq N_k \quad \forall k \in K \tag{4.36}$$

$$\sum_{j\in J_k} END_{sjk}x_{jk} - v_{sk} = 0 \quad \forall s \in S_k \forall k \in K \tag{4.37}$$

$$v_{sk} - MAX_{sk}u_{sk} \leq 0 \quad \forall s \in S_k \forall k \in K \tag{4.38}$$

$$v_{sk} - MIN_{sk}u_{sk} \geq 0 \quad \forall s \in S_k \forall k \in K \tag{4.39}$$

$$x_{jk} \in \{0,1\} \quad \forall j \in J_k, \forall k \in K \tag{4.40}$$

$$y_{gk} \geq 0 \quad \forall g \in G_k, \forall k \in K \tag{4.41}$$

$$v_{sk} \geq 0 \quad \forall s \in S_k \forall k \in K \tag{4.42}$$

$$u_{sk} \in \{0,1\} \quad \forall s \in S_k \forall k \in K \tag{4.43}$$

The maintenance base planning model can be thought of as the aircraft scheduling model with three additional sets of constraints [Equations (4.37) - (4.39)]. Equations (4.37) count v_{sk}, the total number of maintenance checks at each station for each fleet type. The structure of this equation ensures that if the strings have integer values, so will the v_{sk} variables. Equations (4.38) and (4.39) place maximum and minimum limits on the number of maintenance checks at each station over the week. By using the binary variables u_{sk}, we can also decide if a particular station should offer service to a specific fleet or not. Equations (4.42) and (4.43) will ensure the integrality of the v_{sk} and u_{sk} variables respectively.

4.8.3 Earlier Work and Extensions

Our model is an extension of the maintenance base planning model of Feo and Bard [1989]. However, our modeling approach differs as follows:

1. In their work, daily "O-T" flight paths are provided as an input, while these are generated automatically by our model, and are not fixed *a priori*.

2. Unlike their model, our model handles multiple fleets.

Extensions

The second point mentioned above takes on importance when one considers that several fleets may be assigned to the same airport. As an example, the natural limit due to hangar capacity may be on the total maintenance checks for *all* fleets, rather than on each individual fleet. Similarly, the labor contract requirements may specify the minimum number of maintenance checks for the entire set of fleets, and not for individual fleets. This can be modeled by using the additional constraints

$$MIN_s u_s \leq \sum_{k \in K} v_{sk} \leq MAX_s u_s \tag{4.44}$$

where $u_s \in \{0, 1\}$ indicates if maintenance of *any* fleet can be carried out at a station ($u_s = 1$) or not ($u_s = 0$), and MIN_s and MAX_s are the minimum and maximum

number of maintenance checks that can be performed at station s for all fleets.

Another extension could be the disaggregation of the v_{sk} variables by day-of-week. For example, we could create new variables v_{sk}^d that represent the number of maintenance checks of fleet k at station s on day d. These are related to v_{sk} by

$$\sum_{d=1,\ldots,7} v_{sk}^d = v_{sk} \tag{4.45}$$

Upper and lower bounds can be placed on these new variables to indicate the upper and lower limit on maintenance at each station for each fleet for each day. These variables can also be used for manpower planning.

Chapter 5

Summary and Extensions

This chapter summarizes the dissertation and presents possible extensions.

5.1 Summary of Dissertation

We first provide a brief overview of each chapter.

Motivation for this Work

Chapter 1 presents a brief analysis of the U.S. airline industry. Since deregulation, the airlines have operated in a very competitive environment. In order to increase profitability and gain a competitive edge, airlines have been developing and investing in decision support tools to aid their schedule planning. Since planning an airline schedule is an extremely complicated task, we present the simplified schedule planning paradigm (Figure 1-3) that is used by many of the major carriers. This paradigm consists of a series of decision modules that have to be solved to make the flight schedule operational. Specifically, we consider the following decisions (Figure 1-4).

1. Fleet Assignment.

2. Through Flight Assignment.

3. Aircraft Maintenance Routing.

4. Crew Deadhead Selection.

5. Crew Pairing Optimization.

These decisions are typically solved sequentially primarily due to computational limitations. Sequential solution then leads to an iterative decision process, i.e., aircraft and crew schedules have to be repeatedly modified until any operational difficulties have been ironed out. Solving these decision modules simultaneously rather than sequentially yields the following benefits:

1. More economical schedules are achieved; and

2. incompatibilities between the decision modules are reduced. This may lead to indirect benefits, such as greater ease in making the schedule operational, and increased productivity of schedule planners.

Review of Methodology

Many of the mathematical formulations used to model the aircraft scheduling decision modules are large. They require the development of sophisticated solution procedures. Chapter 2 presents the methods, namely *branch-and-bound*, *branch-and-price*, and *column generation*, that we use to solve large integer problems of these types.

Crew Scheduling

Chapter 3 presents the crew pairing optimization problem for a long haul airline and provides an extensive survey of earlier work on this problem. It also presents a model based on duty periods, that closely approximates the crew pairing problem. The duty based model can be used to select deadheads efficiently and reduce crew costs. The information generated by this model can also be used to speed up exact solution of the crew pairing problem. Computational results using data from a real long haul airline show that our new solution approach can improve pay-and-credit costs by 20% and run times by 80% over traditional approaches.

Aircraft Scheduling

Chapter 4 presents the fleet assignment problem, the through flight assignment problem, and the aircraft maintenance routing problem for long-haul airlines, and briefly describes earlier work on each of these three decision modules. Using data from a long haul airline, we show that the sequential solution of these decision modules may not always yield feasible maintenance schedules. We develop an aircraft scheduling model, i.e., the integrated fleet assignment, through flight assignment, and aircraft maintenance routing model. Our aircraft scheduling model uses variables, defined as sequences of flights, called strings, that start and end at maintenance stations. Since the number of possible variables in a problem can be potentially large, a branch-and-price algorithm is used to solve this problem. Using the same long haul data, we show that good integer solutions can be obtained in reasonable amounts of time.

We extend the aircraft scheduling model to include maintenance base planning considerations, and show how it can be used to decide which maintenance stations to open or close, which fleets to maintain at each station, and how many maintenance workers should be staffed at each station on each day of the week.

Conclusions

In the past, the decision modules of schedule planning were typically solved sequentially due to computational and algorithmic limitations. In this dissertation, we have shown that simultaneous solutions are not just beneficial, but also possible with state-of-the-art modeling, algorithmic, and computational capabilities. In the next section, we extend our new models by further integration and discuss their potential benefits and tractability.

5.2 Extensions

There are various extensions to our work on aircraft and crew scheduling.

5.2.1 Crew Pairing Extensions

The following sections highlight potential future work in the area of crew pairing optimization.

Model Extensions

Some crew planning models have additional crew restrictions [Barnhart et al 1993]. Such constraints can be added to the crew pairing formulation [Equations (3.8) - (3.10)]. Other model extensions can be introduced to capture more complex rules such as limitations on total flying time in a pairing, constraints on flying per day, and so on [Barnhart et al 1993].

Evaluation of Recovery Models

Models are used by many airlines to reschedule its operations after a disruption such as weather or mechanical breakdown renders the current plan inoperable [Jarrah et al 1993]. Since our duty period model generates quick bounds to the crew pairing problem, it can be used to give quick estimates of the best possible cost of a new plan/schedule. Operational replanning and rescheduling models can be evaluated against these benchmarks.

5.2.2 Aircraft Scheduling Extensions

Various extensions can be made to the aircraft scheduling model to capture more operational or business constraints.

Extensions to Fleet Assignment

While the fleet assignment model [Equations (4.7) - (4.14)] is able to provide a minimum number of maintenance opportunities without destroying tractability, there is no consideration of the effects of the fleeting decision on crew pairings. While it is difficult to explicitly consider the effects of fleeting on crew pairing costs, one can add constraints such as *elimination of lonely overnights* and *mid-day breakthroughs*

[Clarke et al 1995b] to ensure that the crew pairings are not too expensive. Another method that could be employed is the addition of constraints that limit the total flying hours for each fleet type by the total flying hours available for the crew assigned to that fleet [Clarke et al 1995b].

Extensions to Through Assignment

One of the assumptions made in the long-haul aircraft scheduling model is that there are not many through itineraries, offered by the same airline, that compete for the same origin-destination market in international travel. If this assumption is invalid, i.e., there are a number of competing itineraries for a given market, then constraints have to be added that ensure that only one itinerary is considered [Talluri and Gopalan 1995].

Extensions to Aircraft Routing

Many airlines require that the aircraft rotations that satisfy maintenance requirements also satisfy an Euler tour requirement, i.e., every aircraft in a fleet is assigned to fly a rotation that covers each flight assigned to that fleet [Clarke et al 1995]. The modification of the aircraft scheduling problem to handle this requirement is not trivial since it does not explicitly consider rotations, but strings. The addition of this requirement would require the generation of subtour elimination cuts [Clarke et al 1995].

Solving Aircraft Scheduling

In the solution of the aircraft scheduling problem, some parameters were fixed. For example, the number of non-integer follow-ons and non-integer fleet-flight pairs fixed at each node were limited to 20 and 5 respectively. While these numbers were arrived at by some preliminary tests, one could try a variety of scenarios to test the effect of these parameters on the solution. Another aspect of the solution is that we did not use *backtracking* in the branch-and-price algorithm [Bradley et al 1977, Barnhart

et al 1995b]. Backtracking is computationally intensive and non-trivial. However, it may be essential in certain cases to find a good solution.

5.2.3 Integrated Aircraft and Crew Scheduling

Crew costs are among the largest operating costs faced by an airline (Table 1.3) and crews are usually assigned to fly a specific fleet type[1]. On the other hand, aircraft scheduling decisions have a huge impact on revenues. While it would seem natural that crew and aircraft scheduling decisions be made simultaneously, to the best of our knowledge, there has been no work on the consideration of crew costs in fleet assignment decisions. Even the work of Clarke et al [1995b] attempts to provide better crew connections, but does not explicitly consider costs. We present models that attempt to combine these decision modules into one integrated model.

Challenges of Integrated Modeling

The integrated crew and aircraft scheduling model will be difficult to solve for the following reasons:

1. Even sequentially, the crew pairing problem and the aircraft scheduling problem are hard problems to solve. The integrated model would likely be intractable.

2. The crew pairing problem is difficult to solve even though it is solved over only the flights in a crew compatible fleet group. In an integrated model, since the decision of assigning flights to fleets is made simultaneously with the decision of assigning flights to pairings, each crew pairing problem will have to be solved over *all* flights in the schedule. This would dramatically increase the complexity and tractability of the integrated model.

[1]Actually, there are groups of fleets, called *crew compatible* groups, which have the same crew requirement but different seating arrangements. Therefore, while each fleet in such groups has to be considered separately for marketing reasons, they can be considered the same for crew scheduling purposes [Clarke et al 1995b]. Such groups are also called *pilot aggregates* [Subramanian et al 1994].

Use of Approximations

Since it seems impractical at this point to solve the integrated aircraft and crew scheduling problems, we take the approach of providing a *reasonable approximation* to this problem. We build on the following observations made in chapters 3 and 4.

1. The linear relaxation of the duty period formulation (DPP) [Equations (3.11) - (3.15)] gives tight bounds and useful information about the crew pairing problem (CPP) [Equations (3.8) - (3.10)] in about half a minute.

2. The linear relaxation of the enhanced fleet assignment problem [Equations (4.7) - (4.14)] yields tight bounds to the aircraft scheduling model [Equations (4.15)-(4.20)] in about a sixth of the time to solve the aircraft scheduling LP.

In summary, we observe that the approximate models give tight bounds to exact *large integer programs.* Using this, we develop our integrated model that combines DPP-IP with the fleet assignment IP, instead of CPP-IP with the aircraft scheduling IP.

Notation for Approximate Integrated Model

The integrated model combines the fleet assignment model [Equations (4.7) - (4.14)] with the duty period formulation [Equations (3.11) - (3.15)]. In order to prevent confusion, the DPP variables are all changed to uppercase, the crew base index in DPP is changed from k to R to prevent confusion with the fleet index, and an extra subscript K is added to identify the fleet to which a crew is assigned.

The cost of assigning flight i to fleet k is e_{ik} (section 4.2.1). Binary variable z_{ik} (z^m_{ik}) takes on a value of 1 if flight arc (maintenance arc) variable i is assigned to fleet k, and 0 otherwise (section 4.2.4). The cost of assigning a crew[2] from base R assigned to fleet K to fly duty J is C^{RK}_J, and X^{RK}_J is equal to 1 if crew from base R assigned to fleet K is assigned to duty J. The ground arc variables in the fleet assignment and duty formulations are represented by the variables w_{hk} and Y^{RK}_J respectively.

[2]Recall from chapter 3 that the cost of a duty is assumed to be the Time-Away-From-Base (TAFB) costs.

Coefficients δ_{pik} and δ^m_{pik} are equal to 1 if $p = i$ and 0 otherwise. Coefficient A^K_{IJ} is equal to 1 if duty J assigned to fleet K covers flight I and 0 otherwise. Coefficient $b3_{nik}$ ($b3^m_{nik}$) is equal to 1 if flight arc (maintenance arc) i in fleet k ends at node n, -1 if it begins at node n, and 0 otherwise. Coefficients $b4_{nhk}$ and B^K_{LJ} are similarly defined for the fleet assignment ground arcs and for the duty period arcs respectively. In order to count the aircraft, a *count time* is arbitrarily chosen, and the total flow on the fleet assignment arcs at this time is counted. Coefficient $d3_{ik}$ ($d3^m_{ik}$) take on value 1 if the flight arc (maintenance arc) i for fleet k crosses the count line, and 0 otherwise. The coefficient $d4_{hk}$ is similarly defined for the fleet assignment ground arcs.

The Approximate Integrated Model

$$\min \sum_k \sum_i e_{ik} z_{ik} + \sum_k \sum_i e_{ik} z^m_{ik} + \sum_K \sum_R \sum_J C^{RK}_J X^{RK}_J + \sum_K \sum_R \sum_J D^{RK}_J Y^{RK}_J \tag{5.1}$$

$s.t.$

$$\sum_k \sum_i \delta_{pik} z_{ik} + \sum_k \sum_i \delta^m_{pik} z^m_{ik} = 1 \; \forall p \tag{5.2}$$

$$\sum_i b3_{nik} z_{ik} + \sum_i b3^m_{nik} z^m_{ik} + \sum_h b4_{nhk} w_{hk} = 0 \; \forall n, \forall k \tag{5.3}$$

$$\sum_i d3_{ik} z_{ik} + \sum_i d3^m_{ik} z^m_{ik} + \sum_h d4_{hk} w_{hk} \leq N_k \; \forall k \tag{5.4}$$

$$\sum_{i \in R^4_t} z^m_{ik} \geq N_k \; \forall t, \forall k \tag{5.5}$$

$$\sum_i z^m_{ik} \geq \left\lceil \frac{7N_k}{4} \right\rceil \forall k \in K \tag{5.6}$$

$$\sum_i (-\delta_{pik}) z_{ik} + \sum_i (-\delta_{pik}) z^m_{ik} + \sum_R \sum_J A^k_{iJ} X^{Rk}_J \geq 0 \; \forall i, \forall k \tag{5.7}$$

$$\sum_J B^K_{LJ} X^{RK}_J + \sum_J B^K_{LJ} Y^{RK}_J = 0 \; \forall L, \forall R, \forall K \tag{5.8}$$

$$z_{ik}, z^m_{ik} \in \{0, 1\} \; \forall i, \; \forall k \tag{5.9}$$

$$w_{hk} \geq 0 \; \forall h, \; \forall k \tag{5.10}$$

$$X^{RK}_J \in \{0, 1\} \; \forall J, \forall R, \forall K \tag{5.11}$$

$$Y^{RK}_J \geq 0 \; \forall J, \forall R, \forall K \tag{5.12}$$

The objective function (5.1) of the integrated model minimizes the sum of the total fleet assignment costs and total TAFB duty period costs. Flight cover contraints (5.2), one for each flight, ensure that all the flights in the schedule are assigned to exactly one fleet. Note that these are the only cover constraints in the integrated model.

Balance constraints (5.3), one set for each fleet type, ensure that balance is maintained for each fleet at each station at each point in time. Count constraints (5.4), one for each fleet type, ensure that the number of aircraft used in each fleet do not exceed the number available in that fleet. Valid inequalities (5.5) and (5.6) ensure that the minimum number of maintenance opportunities are provided by the fleet assignment model (section 4.2.4).

The binding constraints (5.7) ensure that if a flight is assigned to a fleet type, it has to be flown by a crew member who is also assigned to that fleet type. There is one such constraint for each flight for each fleet type. Note that this is the only constraint that ensures flow on the duties. There is no flight cover constraints on the duties.

Balance constraints (5.8), one set for each fleet for each crew base, ensure that balance is maintained for each crew at each station for each crew base for each fleet type. Constraints (5.9) and (5.10) ensure the integrality of the fleet assignment variables, and (5.11) and (5.12) ensure the integrality of the duty formulation variables.

Model Depiction

An example of the constraint matrix for the integrated model is depicted in Figure 5-1. In this example, there are two fleet groups. The crew assigned to the first group have three crew bases, and the crew in the second group have two bases (not necessarily different from the first). The constraints are desribed below (for simplicity, the fleet assignment maintenance contraints, (5.5) and (5.6) are not shown):

- Constraints (A), (B), and (C) belong to the fleet assignment model, and correspond to constraints (5.2), (5.3), and (5.4) respectively.

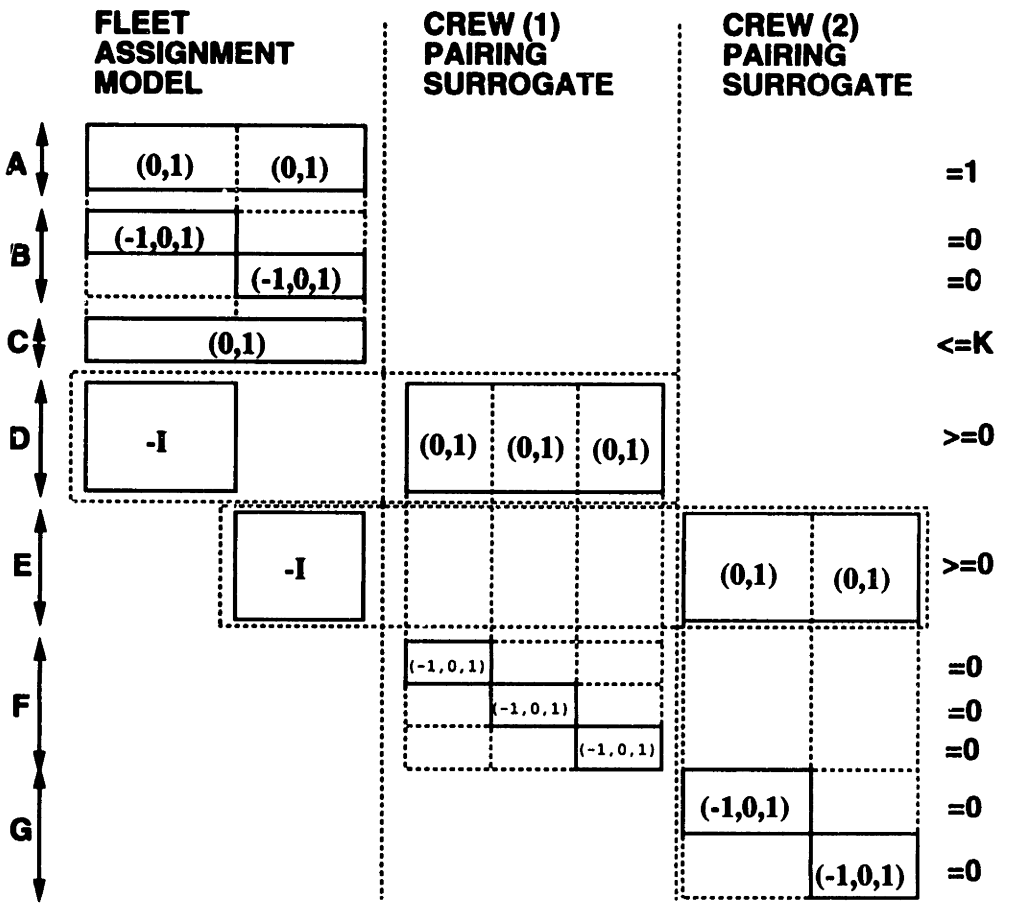

Figure 5-1: Combined Scheduling of Aircraft and Crew

- Constraints (D) and (E) are the binding constraints corresponding to the constraints (5.7).
- Constraints (F) and (G) are crew balance constraints corresponding to the (5.8).

Model Properties

Since the individual models (duty period and fleet assignment models) give tight bounds, we expect that the integrated model [Equations (5.1) - (5.12)] will give good bounds on the scheduling problem. However, the integrated model is a large integer program. Although it can be solved without column generation, it can still be difficult to solve due to the large number of constraints. Recall from section 4.7.2 that the number of constraints in the fleet assignment model was 7,155 constraints. In addition, the integrated model has one binding constraint for each flight for each fleet type. Given 1,124 flights and 9 fleet types, this amounts to 10,116 constraints. Including the duty period balance constraints, the total number of constraints in the integrated model would be about 30,000 for the problem we considered in chapter 4 [Barnhart 1996]. However, models of this magnitude have been solved [Hane et al 1994]. One simplification could be to relax the integer constraints on the variables. This could be done either for all the variables or for subsets of variables. As in section 4.7.2, the linear solutions can be used to aid in the solution of the integer problem. Moreover, the linear relaxations can give good bounds against which solutions can be evaluated.

Extensions

Another extension would be to replace the fleet assignment model with the aircraft scheduling model [Equations (4.15)-(4.20)]. The new integrated model is reproduced below. The notation is similar to the previous model. The cost of string j in fleet k is c_{jk}. String variable x_{jk} equals 1 if string j is assigned to fleet k and 0 otherwise. Coefficient a_{ijk} equals 1 if string j in fleet k covers flights i and 0 otherwise. The balance and count coefficients for the aircraft scheduling model are similar to those

in the fleet assignment model.

$$
\begin{aligned}
\min & \sum_k \sum_j c_{jk} x_{jk} + \\
& \sum_K \sum_R \sum_J C_J^{RK} X_J^{RK} + \sum_K \sum_R \sum_J D_J^{RK} Y_J^{RK} && (5.13) \\
s.t. & \\
& \sum_k \sum_j a_{ijk} x_{jk} = 1 \; \forall i && (5.14) \\
& \sum_j b1_{ljk} x_{jk} + \sum_g b2_{lgk} y_{gk} = 0 \; \forall l, \forall k && (5.15) \\
& \sum_j d1_{jk} x_{jk} + \sum_g d2_{gk} y_{gk} \leq N_k \; \forall k && (5.16) \\
& \sum_i (-a_{ijk}) x_{jk} + \sum_R \sum_J A_{iJ}^k X_J^{Rk} \geq 0 \; \forall i, \forall k && (5.17) \\
& \sum_J B_{LJ}^K X_J^{RK} + \sum_J B_{LJ}^K Y_J^{RK} = 0 \; \forall L, \forall R, \forall K && (5.18) \\
& x_{jk} \in \{0,1\} \; \forall j, \forall k && (5.19) \\
& y_{gk} \geq 0 \; \forall g, \forall k && (5.20) \\
& X_J^{RK} \in \{0,1\} \; \forall J, \forall R, \forall K && (5.21) \\
& Y_J^{RK} \geq 0 \; \forall J, \forall R, \forall K && (5.22)
\end{aligned}
$$

The objective function (5.13) minimizes the total cost of strings and total duty TAFB costs. Cover constraints (5.14) ensure that each flight is assigned to one string, and hence one fleet. Balance constraints (5.15) ensure balance between strings and ground arcs at each station for each fleet at each point in time. Constraints (5.16) ensure that the number of aircraft used do not exceed the number available.

The binding constraints in this problem are (5.17) and ensure that if a flight is assigned to a string of a particular fleet, the crew that will fly that flight should also be assigned to fly that fleet. Constraints (5.19) and (5.20) ensure integrality of the string and ground arc variables. The rest of the constraints are similar to the previous integrated model.

Although this model will be harder to solve, it will ensure the satisfaction of maintenance constraints instead of just providing a lower bound on the optimal solution

value (like with the fleet assignment model). It will also have fewer constraints (because there are no minimum maintenance opportunity contraints). The solution of this model will be achieved in a similar manner to that of the aircraft scheduling model (section 4.6) except that the reduced cost of a string is

$$\bar{c_{jk}} = c_{jk} - \sum_{i \in S_{jk}} \pi_i + \lambda^s_{jk} - \lambda^e_{jk} - d1_{jk}\sigma_k + \sum_{i \in S_{jk}} \pi^B_{ik} \tag{5.23}$$

where π^B_{ik} is the dual variable associated with the binding constraint for flight i in fleet k.

Another possible formulation results by replacing the duty period formulation with a crew pairing formulation.

The Combined Crew and Aircraft Scheduling Model

In this section, we combine the set covering version of the crew pairing model [Equations (3.8) - (3.10)] with the aircraft scheduling model [Equations (4.15) - (4.20)]. As before, the variables in the crew scheduling model have been changed to uppercase. In addition, each crew pairing variable is assigned a superscript K that denotes which fleet the crew is be assigned to. Note that these variables no longer have the crew base index R because the pairing variables eliminate the need for balance constraints since they start and end at the same crewbase.

$$\min \sum_k \sum_j c_{jk} x_{jk} + \sum_K \sum_J C^K_J W^K_J \tag{5.24}$$

s.t.

$$\sum_k \sum_j a_{ijk} x_{jk} = 1 \; \forall i \tag{5.25}$$

$$\sum_j b1_{ljk} x_{jk} + \sum_g b2_{lgk} y_{gk} = 0 \; \forall l, \forall k \tag{5.26}$$

$$\sum_j d1_{jk} x_{jk} + \sum_g d2_{gk} y_{gk} \leq N_k \; \forall k \tag{5.27}$$

$$\sum_i (-a_{ijk}) x_{jk} + \sum_J A^k_{iJ} W^k_J \geq 0 \; \forall i, \forall k \tag{5.28}$$

$$x_{jk} \in \{0,1\}\ \forall j, \forall k \tag{5.29}$$

$$y_{gk} \geq 0\ \forall g, \forall k \tag{5.30}$$

$$W_J^K \in \{0,1\}\ \forall J, \forall K \tag{5.31}$$

The objective function (5.24) minimizes the total cost of strings and pairings. Flight cover constraints (5.25) ensure that each flight in the schedule has to be covered by exactly one string. Balance constraints (5.26) and count constraints (5.27) are as defined earlier. Constraints (5.29) - (5.31) ensure the integrality of the variables.

The binding constraint (5.28) ensures that if a flight is assigned to a fleet, it will be flown by a crew assigned to the same fleet. By allowing the constraint to be an inequality, we allow crew to deadhead on this flight. In this way, we ensure that each flight can be assigned to multiple (deadheading) crew while still being assigned to exactly one fleet type.

This problem will be much harder to solve for the reasons mentioned earlier. Consequently, we recommend that future research begin with the development of solution methods for the approximate integrated model [Equations (5.1) - (5.12)].

References

[1] ABARA, J. Applying Iteger Linear Programming to the Fleet Assignment Problem. *Interfaces 19*, 4 (July-August 1989), 20–28.

[2] AGARD, J. Optimal Selection of Aircraft. In *10th AGIFORS Symposium* (Sydney, Austrailia, November 1970).

[3] AHUJA, R. K., MAGNANTI, T. L., AND ORLIN, J. B. *Network Flows: Theory, Algorithms and Applications.* Prentice Hall Inc., Englewood Cliffs, New Jersey, 1993.

[4] Fleet-Aircraft Maintenance Procedures. Published on the World Wide Web, URL: http://www.amrcorp.com/aa_home/servinfo/maint_pro.htm, 1996. Web site owned and maintained by AMR Corp, Inc.

[5] ANBIL, R., BARNHART, C., HATAY, L., JOHNSON, E. L., AND RAMAKRISHNAN, V. Crew-Pairing Optimization at American Airlines Decision Technologies. In *Optimization in Industry*, T. Ciriani and R. Leachman, Eds. John Wiley & Sons Ltd., 1993, pp. 31–36.

[6] ANBIL, R., GELMAN, E., PATTY, B., AND TANGA, R. Recent Advances in Crew-Pairing Optimization at American Airlines. *Interfaces 21*, 1 (January-February 1991), 62–74.

[7] ANBIL, R., TANGA, R., AND JOHNSON, E. A Global Approach to Crew-Pairing Optimization. *IBM Systems Journal 31*, 1 (1992), 71–78.

[8] ANBIL, R., TANGA, R., AND JOHNSON, E. L. A Global Optimization Approach to Crew Scheduling. Tech. Rep. COC-91-05, Georgia Institute of Technology, 1991b.

[9] APPELGREN, L. H. A Column Generation Algorithm for a Ship Scheduling Problem. *Transportation Science 3* (1969), 53–68.

[10] ARABEYRE, J., FEARNLEY, J., STEIGER, F., AND TEATHER, W. The Airline Crew Scheduling Problem: A Survey. *Transportation Science 3* (1969), 140–163.

[11] Air Transport Association Data and Statistics. Published on the World Wide Web URL: http://www.air-transport.org/data/earnings.htm, 1996. Web site

owned and maintained by the Air Transport Association of America. 1301 Pennsylvania Avenue, NW, Suite 1100, Washington, DC 20004-1707 USA.

[12] ATAMTURK, A., NEMHAUSER, G., AND SAVELSBERGH, M. A Combined Lagrangian, Linear Programming, and Implication Heuristic for Large-Scale Set Partitioning Problems. Tech. Rep. LEC-95-07, School of Industrial and Systems Engineering, Georgia Institute of Technology, September 1995.

[13] BAKER, E., AND FISHER, M. Computational Results for Very Large Air Crew Scheduling Problems. *OMEGA, The International Journal of Management Science 9*, 6 (1981), 613–618.

[14] BAKER, E. K. Efficient Heuristic Algorithms for the Weighted Set Covering Problem. *Computers and Operations Research 8*, 4 (1981b), 303–310.

[15] BAKER, E. K., BODIN, L. D., FINNEGAN, W. F., AND PONDER, R. J. Efficient Heuristic Solutions to an Airline Crew Scheduling Problem. *AIIE Transactions 11*, 2 (June 1979), 79–85.

[16] BALAKRISHNAN, A., CHIEN, W. T., AND WONG, R. T. Selecting Aircraft Routes for Long-Haul Operations: A Formulation and Solution Method. *Transportation Research-B 24B*, 1 (1990), 57–72.

[17] BALL, M., BODIN, L., AND DIAL, R. A Matching Based Heuristic for Scheduling Mass Transit Crews and Vehicles. *Transportation Science 17*, 1 (February 1983), 4–31.

[18] BALL, M., AND MAGAZINE, M. The Design and Analysis of Heuristics. *Networks 11* (1981), 215–219.

[19] BALL, M., AND ROBERTS, A. A Graph Partitioning Approach to Airline Crew Scheduling. *Transportation Science 19*, 2 (May 1985), 107–126.

[20] BANFE, C. F. *Airline Management.* Prentice Hall Inc., Englewood Cliffs, New Jersey, 1992.

[21] BARD, J. F., AND CUNNINGHAM, I. G. Improving Through-Flight Schedules. *IIE Transactions 19*, 3 (September 1987).

[22] BARNHART, C., 1996. Personal communication.

[23] BARNHART, C., HATAY, L., AND JOHNSON, E. L. Deadhead Selection for the Long Haul Crew Pairing Problem. Tech. Rep. COC-91-02, Georgia Institute of Technology, 1991b.

[24] BARNHART, C., HATAY, L., AND JOHNSON, E. L. Deadhead Selection for the Long-Haul Crew Pairing Problem. *Operations Research 43*, 3 (May-June 1995).

[25] BARNHART, C., JOHNSON, E. L., ANBIL, R., AND HATAY, L. A Column Generation Technique for the Long-Haul Crew Assignment Problem. Tech. Rep. COC-91-01, Georgia Institute of Technology, 1991.

[26] BARNHART, C., JOHNSON, E. L., ANBIL, R., AND HATAY, L. A Column Generation Technique for the Long-Haul Crew Assignment Problem. In *Optimization in Industry, Volume II*, T. Ciriani and R. Leachman, Eds. John Wiley and Son, 1993.

[27] BARNHART, C., JOHNSON, E. L., HANE, C. A., AND SIGISMONDI, G. An Alternate Formulation and Solution Strategy for Multi-Commodity Network Flow Problems. Tech. Rep. COC-9102, Georgia Institute of Technology, 1991c.

[28] BARNHART, C., JOHNSON, E. L., NEMHAUSER, G. L., SAVELSBERGH, M. W., AND VANCE, P. H. Branch-and-Price: Column Generation for Solving Huge Integer Programs. Tech. Rep. LEC-94-03, School of Industrial and Systems Engineering, Georgia Institute of Technology, May 1995b.

[29] BARNHART, C., JOHNSON, E. L., NEMHAUSER, G. L., SIGISMONDI, G., AND VANCE, P. Formulating a Mixed Integer Distribution Problem to Improve Solvability. *Operations Research 41*, 6 (November-December 1993b), 1013–1019.

[30] BARNHART, C., AND SHENOI, R. An Alternate Model and Solution Approach for the Long-Haul Crew Pairing Problem. Tech. rep., Massachusetts Institute of Technology, 1996.

[31] BARTLETT, T. An Algorithm For The Minimum Number of Transport Units To Maintain a Fixed Schedule. *Naval Research Logistics Quarterly 4* (1957), 139–149.

[32] BARUTT, J., AND HULL, T. Airline Crew Scheduling: Supercomputers and Algorithms. *SIAM NEWS* (November 1990).

[33] BEASLEY, J., AND CAO, B. An Algorithm for the Crew Scheduling Problem. Unpublished document, April 1993.

[34] BELGRAY, D. C. Discussion of : Optimal Selection of Aircraft by J. Agard. In *10th AGIFORS Symposium* (Sydney, Austrailia, November 1970). David C. Belgray worked for Eastern Airlines.

[35] BELOBABA, P. P. Airline Yield Management: An Overview of Seat Inventory Control. *Transportation Science 21*, 2 (May 1987), 63–73.

[36] BEN-AKIVA, M., AND LERMAN, S. R. *Discrete Choice Analysis: Theory and Application to Travel Demand.* MIT Press, Cambridge, Massachusetts, 1985.

[37] BERGE, C. *Graphs and Hypergraphs.* Elsevier Publishing Co., Inc., New York, 1973.

[38] BERGE, M. E., AND HOPPERSTAD, C. A. Demand Driven Dispatch: A Method For Dynamic Aircraft Capacity Assignment, Models And Algorithms. *Operations Research 41*, 1 (January-February 1993), 153–168.

[39] BERTSIMAS, D., AND TSITSIKILIS, J. Introduction to Optimization, Fall 1994. MIT Course Notes.

[40] BIRGE, J. R. Current Trends in Stochastic Programming Computation and Applications. Tech. rep., Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor MI, August 1995.

[41] BIXBY, R. *Version 3.0 of 'Using the CPLEX Callable Library'*. CPLEX Optimization, Inc., Suite 279, 930 Tahoe Blvd., Bldg. 802, Incline Village, NV 89451-9436, 1994.

[42] BOLAND, N., AND CLARKE, L., 1996. Personal communication.

[43] BOROUGHS, D. L. More And More Companies Are Cutting Costs With State-Of-The-Art Videoconferencing Systems. *U.S. News and World Report* (June 5 1995), 46–48.

[44] BRADLEY, S. P., HAX, A., AND MAGNANTI, T. L. *Applied Mathematical Programming*. Addison-Wesley Publishing Company, Reading, Massachusetts, 1977.

[45] BRANNIGAN, M. ValuJet Poised to Open Low-Cost Service Via New York's La Guardia Airport. *The Wall Street Journal* (March 15th, 1996), B3.

[46] BRANNIGAN, M., AND LISSER, E. D. Delta Air May Try New 'Lite' Service. *The Wall Street Journal* (February 9th, 1996), B8.

[47] BROOKS, R. Southeast Journal: Business Fliers See Big Rise in Some Fares. *The Wall Street Journal* (April 12th, 1995), S1.

[48] BULKELEY, W. M. Picture-Phone Marketers Target the Home PC. *The Wall Street Journal* (February 27th, 1996), B1.

[49] CAREY, S. USAir CEO Stephen Wolf Says Airline Needs an Overhaul of Its Cost Structure. *The Wall Street Journal* (April 17th, 1996), A8.

[50] CAREY, S. Air Treaty Could Boost UAL, Lufthana. *The Wall Street Journal* (March 1st, 1996b), A2.

[51] CARLTON, J. Home Work; Down to Business; The Home Office is Quickly Becoming Less Expensive - And More Sophisticated. *The Wall Street Journal* (June 19th, 1995), R32.

[52] CARRARESI, P., GALLO, G., AND ROUSSEAU, J. Relaxation Approaches to Large-Scale Bus Driver Scheduling Problems. *Transportation Research-B 16B*, 5 (1982), 383–397.

[53] CHRISTOFIDES, N., MINGOZZI, A., AND TOTH, P. State-Space Relaxation Procedures for the Computation of Bounds to Routing Problems. *Networks 11* (1981), 145–164.

[54] CLARKE, L. HANE, C.A.AND JOHNSON, E., AND NEMHAUSER, G. Maintenance and Crew Considerations in Fleet Assignment. Tech. rep., School of Industrial and Systems Engineering, Georgia Institute of Technology, May 1995b.

[55] CLARKE, L., JOHNSON, E. L., NEMHAUSER, G. L., AND ZHU, Z. The Aircraft Rotation Problem. Tech. rep., School of Industrial and Systems Engineering, Georgia Institute of Technology, August 1995. To appear in *Mathematics of Industrial Systems.*

[56] COLE, J. International Lease to Buy Airbus and Boeing Planes. *The Wall Street Journal* (March 5th, 1996), A3.

[57] CRAINIC, G. T., AND ROUSSEAU, J.-M. The Column Generation and Airline Crew Scheduling Problem. *INFOR 25*, 2 (1987), 136–151.

[58] DANTZIG, G., AND FULKERSON, D. Minimizing The Number of Tankers To Meet a Fixed Schedule. *Naval Research Logistics Quarterly 1* (1954), 217–222.

[59] DANTZIG, G. B. *Linear Programming and Extensions.* Princeton University Press, Princeton, New Jersey, 1963, ch. 22.

[60] DANTZIG, G. B. *Linear Programming and Extensions.* Princeton University Press, Princeton, New Jersey, 1963.

[61] DANTZIG, G. B., AND WOLFE, P. Decomposition Principles For Linear Programs. *Operations Research 8* (1960), 101–111.

[62] DANTZIG, G. B., AND WOLFE, P. The Decomposition Algorithm For Linear Programs. *Econometrica 29*, 4 (October 1961), 767–778.

[63] DASBURG, J. S. A Taxing Drag on the Airlines. *The Wall Street Journal* (March 21st, 1995), A20.

[64] DASKIN, M. S., AND PANAYOTOPOULOS, N. D. A Lagrangian Relaxation Approach to Assigning Aircraft to Routes in Hub and Spoke Networks. *Transportation Science 23*, 2 (May 1989).

[65] DENARDO, E. V., AND FOX, B. L. Shortest-Route Methods: 1. Reaching, Pruning and Buckets. *Operations Research 27* (1979), 161–196.

[66] DEO, N., AND PANG, C.-Y. Shortest-Path Algorithms:Taxonomy and Annotation. *Networks 14* (1984), 275–323.

[67] DESAULNIERS, G., DESROSIERS, J., DUMAS, Y., MARC, S., RIOUX, B., SOLOMON, M., AND SOUMIS, F. Crew Pairing at Air France. Tech. Rep. G-93-39, GERAD, May 1995.

[68] DESAULNIERS, G., DESROSIERS, J., IOACHIM, I., SOLOMON, M., AND SOUMIS, F. A Unified Framework for Deterministic Time Constrained Vehicle Routing and Crew Scheduling Problems. Tech. Rep. G-94-46, GERAD, September 1994.

[69] DESAULNIERS, G., DESROSIERS, J., SOLOMON, M. M., AND SOUMIS, F. Daily Aircraft Routing and Scheduling. Tech. rep., GERAD, June 1994b.

[70] DESAULNIERS, G., SOUMIS, F., DESROSIERS, J., GELINAS, E., AND SOLOMON, M. M. Weekly Aircraft Routing And Scheduling. In *INFORMS* (New Orleans, Louisiana, Fall 1995b).

[71] DESROCHERS, M., DESROSIERS, J., AND SOLOMON, M. A New Optimization Algorithm For The Vehicle Routing Problem With Time Windows. *Operations Research 40*, 2 (March-April 1992), 342–354.

[72] DESROCHERS, M., AND SOUMIS, F. A Generalised Permanent Labelling Algorithm For The Shortest Path Problem With Time Windows. *INFOR 26*, 1 (1988), 191–212.

[73] DESROCHERS, M., AND SOUMIS, F. A Reoptimization Algorithm For The Shortest Path Problem With Time Windows. *European Journal of Operations Research 35* (1988b), 242–254.

[74] DESROCHERS, M., AND SOUMIS, F. A Column Generation Approach To The Urban Transit Crew Scheduling Problem. *Transportation Science 23*, 1 (Feb 1989), 1–13.

[75] DESROSIERS, J., DUMAS, Y., DESROCHERS, M., SOUMIS, F., SANSO, B., AND TRUDEAU, P. A Breakthrough in Airline Crew Scheduling. Tech. Rep. G-91-11, GERAD, March 1991.

[76] DESROSIERS, J., DUMAS, Y., SOLOMON, M., AND SOUMIS, F. Time Constrained Routing and Scheduling. Tech. Rep. G-92-42, GERAD, 1992.

[77] DESROSIERS, J., SOUMIS, F., AND DESROCHERS, M. Routing with Time Windows by Column Generation. *Networks 14* (1984), 545–565.

[78] DESROSIERS, J., SOUMIS, F., DESROCHERS, M., AND SAUVE, M. Methods for Routing with Time Windows. *European Journal of Operational Research 23* (1986), 236–245.

[79] DIJKSTRA, E. A Note on Two Problems in Connexion with Graphs. *Numerische Mathematik 1* (1959), 269–271.

[80] DILLON, J., GOPALAN, R., RAMACHANDRAN, S., RUSHMEIR, R., AND TALLURI, K. Fleet Assignment at USAir. In *TIMS/ORSA Joint National Meeting* (Phoenix, Arizona, 1993).

[81] DRUCKERMAN, J., SILVERMAN, D., AND VIARAPULOS, K. Optimization Subroutine Library, Guide and Reference, Release 2, July 1991. Document Number SC23-0519-2, IBM, Kingston, NY.

[82] DUMAS, Y., DESROSIERS, J., AND SOUMIS, F. The Pickup and Delivery Problem with Time Windows. *European Journal of Operational Research 54* (1991), 7–22.

[83] ELCE, I. The Development and Implementation of Air Canada's Long Range Planning Model. In *10th AGIFORS Symposium* (Sydney, Austrailia, November 1970). Ivan Elce worked for Air Canada.

[84] ETSCHMAIER, M. M., AND MATHAISEL, D. F. X. Airline Scheduling : An Overview. *Transportation Science 19*, 2 (1985), 127–138.

[85] FAMIGHETTI, R., Ed. *The World Almanac and Book of Facts.* Richard W. Eiger, Funk and Wagnalls Corporation, One International Boulevard, Suite 444, Mahwah, New Jersey 07495, 1995.

[86] FARKAS, A. *The Influence Of Network Effects And Yield Management On Airline Fleet Assignment Decisions.* PhD thesis, Massachusetts Institute of Technology, 1996.

[87] FARLEY, A. A Note on Bounding a Class of Linear Programming Problems, Including Cutting Stock Problems. *Operations Research 38*, 5 (September-October 1990), 922–924.

[88] FEO, T. A., AND BARD, J. F. Flight Scheduling and Maintenance Base Planning. *Management Science 35*, 12 (December 1989), 1415–1432.

[89] FISHER, M. L. The Lagrangian Relaxation Method for Solving Integer Programming Problems. *Management Science 27*, 1 (January 1981), 1–18.

[90] FLINT, P. Fare's Fair. *Air Transport World* (November 1995), 7.

[91] FLINT, P. A Fare Solution. *Air Transport World* (November 1995b), 61–62.

[92] FORREST, J. J., AND GOLDFARB, D. Steepest-Edge Simplex Algorithms for Linear Programming. *Mathematical Programming 57* (1992), 341–374.

[93] Fuel Cost and Consumption Tables. Published in *Air Transport World*, 1993-1996.

[94] GAMACHE, M., SOUMIS, F., MARQUIS, G., AND DESROSIERS, J. A Column Generation Approach for Large Scale Aircrew Rostering Problems. Tech. rep., GERAD, June 1994.

[95] GAREY, M. R., AND JOHNSON, D. S. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W.H.Freeman and Company, San Fransisco, 1979.

[96] GARFINKEL, R., AND NEMHAUSER, G. The Set-Partitioning Problem: Set Covering with Equality Constraints. *Operations Research 17* (1969), 848–856.

[97] GERBRACHT, R. J. A New Algorithm for Very Large Crew Pairing Problems. In *18th AGIFORS Symposium* (Vancouver, British Columbia, Canada, September 1978), pp. 315–341.

[98] GERSHKOFF, I. Optimizing Flight Crew Schedules. *Interfaces 19*, 4 (July-August 1989), 29–43.

[99] GERTSBAKH, I., AND STERN, H. Minimal Resources for Fixed and Variable Job Schedules. *Operations Research 26*, 1 (January-February 1988), 68–85.

[100] GILMORE, P., AND GOMORY, R. A Linear Programming Approach to the Cutting Stock Problem. *Operations Research 9* (1961), 849–869.

[101] GILMORE, P., AND GOMORY, R. A Linear Programming Approach to the Cutting Stock Problem, Part II. *Operations Research 11* (1963), 863–888.

[102] GIRARD, D. The Airlines Operations Model: A Schedule Development and Evaluation Tool. In *13th AGIFORS Symposium* (Acapulco, Mexico, September 1973), pp. 195–209.

[103] GOLDEN, B., DEARMON, J., AND BAKER, E. Computational Experiments with Algorithms for a Class of Routing Problems. *Computers and Operations Research 10*, 1 (1983), 47–59.

[104] GOLDFARB, D., AND REID, J. A Practicable Steepest-Edge Simplex Algorithm. *Mathematical Programming 12* (1977), 361–371.

[105] GOPALAN, R., AND TALLURI, K. The Aircraft Maintenance Routing Problem. Tech. rep., USAir, Operations Research Department, 2345 Crystal Drive, Arlington, VA 22227, 1995.

[106] GRAVES, G. W., MCBRIDE, R. D., GERSHKOFF, I., ANDERSON, D., AND MAHIDHARA, D. Flight Crew Scheduling. *Management Science 39*, 6 (June 1993), 736–745.

[107] GRAY, D. A. Airworthy: Decision Support for Aircraft Overhaul Maintenance Planning. *OR/MS Today* (December 1992), 24–29.

[108] GRAY, D. A. Consulting Concepts Come to Life For Author While Working on American Airlines Maintenance Project. *OR/MS Today* (December 1993), 22–28.

[109] GRAY, D. A., AND KABBANI, N. Right Tool, Place, Time. *OR/MS Today* (April 1994), 34–41.

[110] GRULEY, B., AND MCCARTNEY, S. Flock of New Low-Fare Carriers Means Savings for Consumers, U.S. Study Says. *The Wall Street Journal* (November 2nd, 1995), A4.

[111] GU, Z., JOHNSON, E. L., NEMHAUSER, G. L., AND WANG, Y. Some Properties Of The Fleet Assignment Problem. *Operations Research Letters 15* (1994), 59–71.

[112] HALL, R. W. Configuration of an Overnight Package Air Network. *Transportation Research-A 23A*, 2 (1989), 139–149.

[113] HANE, C. A., BARNHART, C., JOHNSON, E. L., MARSTEN, R. E., NEMHAUSER, G. L., AND SIGISMONDI, G. The Fleet Assignment Problem: Solving a Large-Scale Integer Program. Tech. Rep. COC-92-04, School of Industrial and Systems Engineering, Georgia Institute of Technology, June 1994.

[114] HAOUARI, M., DEJAX, P., AND DESROCHERS, M. Modelling and Solving Complex Vehicle Routing Problems using Column Generation. Tech. Rep. G-90-22, Les cahiers du GERAD, May 1990.

[115] HARA, M., AND KOYAMA, R. Short Term Planning Model for Domestic Operation. In *13th AGIFORS Symposium* (Acapulco, Mexico, October 1973), pp. 183–194.

[116] HELD, M., WOLFE, P., AND CROWDER, H. P. Validation of Subgradient Optimization. *Mathematical Programming 6* (1974), 62–88.

[117] HERSH, M., AND LADANY, S. P. Optimal Seat Allocation For Flights With One Intermediate Stop. *Computers and Operations Research 5* (1978), 31–37.

[118] HOFFMAN, K., AND PADBERG, M. Solving Airline Crew Scheduling Problems by Branch-and-Cut. *Management Science 39*, 6 (June 1993), 657–682.

[119] HOUCK JR., D. J., PICARD, J. C., QUEYRANNE, M., AND VEMUGANTI, R. The Travelling Salesman Problem as a Constrained Shortest Path Problem: Theory and Computational Experience. *OPSEARCH 17*, 2&3 (1980), 93–109.

[120] JAFFE, G. Southeast Journal: ValuJet Air Plans Service To Charlotte. *The Wall Street Journal* (January 24th, 1996), S1.

[121] JARRAH, A. I. Z., YU, G., KRISHNAMURTHY, N., AND RAKSHIT, A. A Decision Support Framework for Airline Flight Cancellations and Delays. *Transportation Science 27*, 3 (August 1993), 266–280.

[122] JOHNSON, E. L., 1996. Personal communication.

[123] JONES, R. D. Development Of An Automated Airline Crew Bid Generation System. *Interfaces 19*, 4 (July-August 1989), 44–51.

[124] JOSHI, S., AND MUTHUKRISHNAN, R. Incremental Tools for Aircraft Scheduling. In *33rd AGIFORS Symposium* (Chicago, Illinois, 10-14 October 1993), pp. 131–149.

[125] KABBANI, N. M., AND PATTY, B. W. Aircraft Routing at American Airlines. In *32nd AGIFORS Symposium* (Budapest, Hungary, 4-9 October 1992), pp. 11–28.

[126] KARP, R. M. Reducibility Among Combinatorial Problems. In *Complexity of Computer Computations* (New York, 1972), R. Miller and J. Thatcher, Eds., Plenum Press, pp. 85–104.

[127] KARP, R. M. On the Complexity of Combinatorial Problems. *Networks 5* (1975), 45–68.

[128] KELLEY, A., AND POHL, I. *A Book on C.* The Benjamin/Cummings Publishing Company, 1990.

[129] KERNIGHAN, B. W., AND RITCHIE, D. M. *The C Programming Language.* Prentice Hall, 1988.

[130] KLINCEWICZ, J. G., AND ROSENWEIN, M. B. The Airline Exception Scheduling Problem. *Transportation Science 29*, 1 (February 1995), 4–16.

[131] KOLEN, A. W. J., KAN, A. H. G. R., AND TRIENEKENS, A. W. J. M. Vehicle Routing with Time Windows. *Operations Research 35*, 2 (March-April 1987), 266–273.

[132] KONIG, P. Short Term Aircraft Assignment. In *16th AGIFORS Symposium* (July 1976), pp. 169–188.

[133] LARSON, R. C., AND ODONI, A. R. *Urban Operations Research.* Prentice Hall, Englewoods Cliffs, New Jersey 07632, 1981.

[134] LASDON, L. S. *Optimization Theory for Large Systems.* Macmillan Publishing Co., Inc., New York, 1970.

[135] LAVOIE, S., MINOUX, M., AND ODIER, E. A New Approach for Crew Pairing Problems by Column Generation with an Application to Air Transportation. *European Journal of Operational Research 35* (1988), 45–58.

[136] LEMKE, C., SALKIN, H., AND SPIELBERG, K. Set Covering by Single Branch Enumeration with Linear-Programming Subproblems. *Operations Research 19* (1971), 998–1022.

[137] LISSER, E. D., AND NOMANI, A. Q. Valujet Negotiates With Other Airlines On Taking Over Its Fleet Maintenance. *The Wall Street Journal* (June 10th, 1996), A4.

[138] LOCHAN, R. C. Long Haul Fleet Assignment: Models, Methods and Applications. Master's thesis, Massachusetts Institute of Technology, May 1995.

[139] MAGNANTI, T. L. *Contributions to Operations Research and Economics.* MIT Press, Cambridge, Massachusetts, 1988, ch. 7. Edited by B. Cournet and H. Tulkens.

[140] MAGNANTI, T. L., SHAPIRO, J. F., AND WAGNER, M. Generalized Linear Programming Solves the Dual. *Management Science 22*, 11 (July 1976), 1195–1203.

[141] MARSTEN, R. E., MULLER, M. R., AND KILLION, C. L. Crew Planning At Flying Tiger:A Successful Application Of Integer Programming. *Management Science 25*, 12 (December 1979), 1175–1183.

[142] MARSTEN, R. E., AND SHEPARDSON, F. Exact Solution of Crew Scheduling Problems Using the Set Partitioning Model: Recent Successful Applications. *Networks 11* (1981), 165–177.

[143] MCCARTHY, M. Under Siege: Major Airlines Find Their Fortress Hubs Aren't Impenetrable. *The Wall Street Journal* (February 6th, 1996), A1.

[144] MCCARTHY, M. United Sees Strong Rise In Traffic, While Costs May Also Increase. *The Wall Street Journal* (March 11th, 1996b), B4.

[145] MCCARTHY, M. Traffic Slows in U.S. Airlines' Big Season; Trend May Signal Softness in Discretionary Spending. *The Wall Street Journal* (October 17th, 1995), A2.

[146] MCCARTHY, M., AND LISSER, E. D. United Studies Merging its Reservation System. *The Wall Street Journal* (March 5th, 1996), A3.

[147] MCCARTHY, M., AND MCCARTNEY, S. United, Southwest Results Hurt By Fuel, Expenses in 1st Quarter. *The Wall Street Journal* (April 24th, 1996), B4.

[148] MCCARTNEY, S. Free Airline Miles Become a Potent Tool for Selling Everything. *The Wall Street Journal* (April 15th, 1996), A1.

[149] MCCARTNEY, S. High Hopes: Conditions Are Ideal For Starting an Airline And Many Are Doing It. *The Wall Street Journal* (April 1st, 1996b), A1.

[150] MCCARTNEY, S. Continental Air's First-Period Net Sets Record, Aided By High Fares. *The Wall Street Journal* (April 23rd, 1996c), B4.

[151] MCCARTNEY, S. Airline Alliances to Alter Overseas Travel. *The Wall Street Journal* (June 11th, 1996d), B1.

[152] MCCARTNEY, S. Why Airline Fare Sales Aren't Always a Bargain. *The Wall Street Journal* (March 11th, 1996e), B1.

[153] McCartney, S., and McCarthy, M. AMR, TWA Post Improved Results for First Quarter. *The Wall Street Journal* (April 18th, 1996f), B4.

[154] Michael, S. C., and Silk, A. J. American Airlines' Value Pricing (A). Case Study 9-594-001, Harvard Business School, Boston, MA 02163, May 11 1994.

[155] Michael, S. C., and Silk, A. J. American Airlines' Value Pricing (B). Case Study 9-594-019, Harvard Business School, Boston, MA 02163, May 11 1994b.

[156] Miller, L. Travel Prices are Headed Up, Up and Away. *The Wall Street Journal* (August 4th, 1995), B1.

[157] Miller, L. Stung by Fee Cuts, Travel Agents Declare War on Big Airlines. *The Wall Street Journal* (March 27th, 1995b), B1.

[158] Minoux, M. Column Generation Techniques in Combinatorial Optimization - A New Application to Crew Pairing. In *24th AGIFORS Symposium* (September 1984), pp. 15–29.

[159] Minoux, M. *Mathematical Programming: Theory and Algorithms.* Wiley-Interscience, John Wiley and Sons, 1986.

[160] Nemhauser, G. L., Savelsbergh, M. W., and Sigismondi, G. C. MINTO, a Mixed INTeger Optimizer. *Operations Research Letters 15* (1994), 47–58.

[161] Nomani, A. Q. Eastern Airports Still Out of Reach of Start-Ups. *The Wall Street Journal* (April 25th, 1996), B1.

[162] Nomani, A. Q. Open Skies Pact Propels U.S.-Canada Flights. *The Wall Street Journal* (March 5th, 1996b), A2.

[163] Nomani, A. Q., and Carey, S. British Airways, American Airlines May Forge Powerful Strategic Alliance *The Wall Street Journal* (May 17th, 1996), A2.

[164] Nugent, T. M., Ed. *Aerospace and Air Transport: Basic Analysis.* Industry Surveys. Standard and Poor, July 1st, 1993.

[165] Odier, E., Lascaux, F., and Hie, H. Medium Haul Trip Pairing Optimization. In *23rd AGIFORS symposium* (October 1983), pp. 81–109.

[166] Official Airlines Guides, Inc. *OAG Desktop Flight Guide.* Oak Brook, Illinois, 1991.

[167] Oster, S. M. *Modern Competitive Analysis.* Oxford University Press, 1994, ch. 3: Industry Analysis, pp. 31–48.

[168] Page, P. Rising Jet Fuel Costs Snap Three-Year Spell of Stability; Carriers have no Rate Hike Plans. *Traffic World* (April 15th, 1996), 29–30.

[169] PAPADIMITRIOU, C. H., AND STEIGLITZ, K. *Combinatorial Optimization: Algorithms and Complexity.* Prentice-Hall Inc., New Jersey, 1982.

[170] PARKER, M., AND RYAN, J. A Column Generation Algorithm for Bandwidth Packing. *Telecommunications Systems* (1994). To appear.

[171] PATTERSON, G. A. United, American Offer Discounts, Joining Fare War. *The Wall Street Journal* (April 3rd, 1995), B4.

[172] PHILLIPS, R. L., BOYD, D. W., AND GROSSMAN JR., T. A. An Algorithm For Calculating Consistent Itinerary Flows. *Transportation Science 25*, 3 (August 1991), 225–239.

[173] PORTER, M. E. *Competitive Strategy.* Free Press, 1980, ch. 1: The Structural Analysis of Industries, pp. 3–33.

[174] QUINTANILLA, C. USAir's New Chairman Charts Cost Cutting Course. *The Wall Street Journal* (March 5th, 1996), B6.

[175] QUINTANILLA, C. Northwest Airlines Cuts Holiday Fares as much as 50%. *The Wall Street Journal* (October 13th, 1995), A4.

[176] RANNOU, B. A New Approach to Crew Pairing Optimization. In *26th AGIFORS Symposium* (October 1986), pp. 153–167.

[177] RAPLEY, K. Short Haul Fleet Planning Models. In *15th AGIFORS Symposium* (Rotorua, New Zealand, October 1975), pp. 307–338.

[178] REPORTER, S. Southwest Airlines Offers Special Fares for the Summer. *The Wall Street Journal* (April 15th, 1996), A11C.

[179] REPORTER, S. Northwest Airlines' Net of $53.4 Million Set 1st-Period Record. *The Wall Street Journal* (April 19th, 1996b), A4.

[180] REPORTER, S. Four Airlines Slash Fares for Summer by Up to 50%. *The Wall Street Journal* (April 5th, 1996c), B2.

[181] REPORTER, S. Bandits at Nine O' Clock. *The Economist* (February 17th, 1996), 57–58.

[182] REPORTER, S. Washington's Wake-up Call. *The Economist* (January 20th, 1996b), 61–63.

[183] REPORTER, S. The Accidental Superhighway. *The Economist* (July 1st, 1995). A Survey of the Internet.

[184] REPORTER, S. United Air to Open Two Foreign Bases For Flight Attendants. *The Wall Street Journal* (June 10th, 1996d), A11B.

[185] REPORTER, S. Stretched To The Limit. *The Economist* (June 17th, 1995b), 72–73.

[186] REPORTER, S. Delta Air Lines Cuts Fares Up to 40% for the Spring. *The Wall Street Journal* (March 21st, 1996e), A13.

[187] REPORTER, S. Blue Skies, Red Ink, Black Future. *The Economist* (May 6th, 1995c), 61.

[188] REPORTER, S. Flying in Formation? *The Economist* (November 18th, 1995d), 73.

[189] REPORTER, S. Latin American Airlines: Unite and Fight. *The Economist* (October 21st, 1995e), 72.

[190] REPORTER, S. The Death of Distance. *The Economist* (September 30th, 1995f). A Survey of Telecommunications.

[191] RIBIERO, C. C., MINOUX, M., AND PENNA, M. C. An Optimal Column-Generation-with-Ranking Algorithm for Very Large Set Partitioning Problems in Traffic Assignment. *European Journal of Operational Research 41* (1989), 232–239.

[192] ROCKAFELLER, T. R. *Convex Analysis.* Princeton University Press, Princeton, New Jersey, 1970.

[193] RUBIN, J. A Technique for the Solution of Massive Set Covering Problems, with Applications to Airline Crew Scheduling. *Transportation Science 7*, 1 (February 1973), 31–48.

[194] RYAN, D., AND FOSTER, B. An Integer Programming Approach to Scheduling. In *Computer Scheduling of Public Transport: Urban Passenger Vehicle and Crew Scheduling* (1981), A. Wren, Ed., North Holland Publishing Company, pp. 269–280.

[195] RYAN, D. M., AND FALKNER, J. C. On The Integer Properties of Scheduling Set Partitioning Models. *European Journal of Operational Research 35* (1988), 442–456.

[196] SAPOUNTZIS, T. Using Distributed Rotations to Generate Aircraft Rotations at American Airlines. In *33rd AGIFORS Symposium* (Evanston, Illinois, October 10-14 1993), pp. 67–81.

[197] SAVELSBERGH, M. A Branch-and-Price Algorithm for the Generalized Assignment Problem. Tech. Rep. COC-9302, Georgia Institute of Technology, 1993.

[198] SAVELSBERGH, M. W., AND NEMHAUSER, G. Functional Description of MINTO, a Mixed INTeger Optimizer. Tech. rep., Georgia Institute of Technology, 1996. Version 2.0a.

[199] SCHRAGE, L. Implicit Representation of Variable Upper Bounds in Linear Programming. *Mathematical Programming Study 4* (1975), 118–132.

[200] SHAPIRO, J. F. *Mathematical Programming: Structures and Algorithms.* Wiley-Interscience, John Wiley and Sons, 1979.

[201] SHEFFI, Y., 1996. Personal communication.

[202] SHENOI, R. G. Solving the Long-Haul Crew Pairing Problem. Master's thesis, Massachusetts Institute of Technology, February 1994.

[203] SHEPARDSON, F., AND MARSTEN, R. E. A Lagrangean Relaxation Algorithm for the Two Duty Period Scheduling Problem. *Management Science 26*, 3 (March 1980), 274–281.

[204] SHEPHERD, W., AND BROCK, J. Airlines. In *The Structure of American Industry*. Prentice Hall, 1995, pp. 247–272.

[205] SHERALI, H. D., AND MYERS, D. C. Dual Formulations and Subgradient Optimization Strategies for Linear Programming Relaxations of Mixed-Integer Programs. *Discrete Applied Mathematics 20* (1988), 51–68.

[206] SIMPSON, R. W., AND BELOBABA, P. 16.74, Air Transportation Economics, Fall 1993. Class notes.

[207] SOLOMON, M., AND DESROSIERS, J. Time Window Constrained Routing and Scheduling Problems. *Transportation Science 22*, 1 (February 1988), 1–13.

[208] SOLOMON, M. M. Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints. *Operations Research 35*, 2 (March-April 1987), 254–265.

[209] SOUMIS, F., FERLAND, J. A., AND ROUSSEAU, J. M. A Model for Large-Scale Aircraft Routing and Scheduling Problems. *Transportation Research-B 14B* (1980), 191–201.

[210] Statistical Abstract of the United States: 1995. U.S. Bureau of the Census, Washington, D.C., 1995. (115th Edition.).

[211] SUBRAMANIAN, R., AND MARSTEN, R. E. Strategic Planning and Scheduling of Aircraft Heavy Maintenance at Delta Air Lines. In *33rd AGIFORS Symposium* (Chicago, Illinois, 10-14 October 1993), pp. 151–168.

[212] SUBRAMANIAN, R., SCHEFF JR., R. P., QUILLINAN, J. D., WIPER, D. S., AND MARSTEN, R. E. Coldstart: Fleet Assignment at Delta Air Lines. *Interfaces 24*, 1 (January-February 1994), 104–120.

[213] SULLIVAN, A. Truckers, Airlines Fret Over Fuel Costs. *The Wall Street Journal* (April 23rd, 1996), A2.

[214] TALLURI, K. Swapping Applications in a Daily Airline Fleet Assignment. Tech. rep., USAir, Operations Research Department, 2345 Crystal Drive, Arlington, VA 22227, 1994.

[215] TALLURI, K. The Four-Day Aircraft Maintenance Routing Problem. Tech. rep., USAir, Operations Research Department, 2345 Crystal Drive, Arlington, VA 22227, 1994b.

[216] TALLURI, K. T., AND GOPALAN, R. Mathematical Models in Airline Schedule Planning: A Survey. Tech. rep., USAir, Operations Research Department, 2345 Crystal Drive, Arlington, VA 22227, 1995.

[217] TEDONE, M., AND GRAY, D. A. Decision Support for Aircraft Overhaul Maintenance Planning. In *31st Annual AGIFORS Symposium* (Brainerd, Minnesota, 14-18 October 1991), pp. 205–218.

[218] TEODOROVIC, D., AND GUBERINIC, S. Optimal Dispatching Strategy on an Airline Network After a Schedule Perturbation. *European Journal of Operational Research 15* (1984), 178–182.

[219] THIRIEZ, H. Airline Crew Scheduling:A Group Theoretic Approach. *Working Paper R69-1* (October 1969).

[220] VANCE, P. *Crew Scheduling, Cutting Stock, and Column Generation:Solving Huge Integer Programs.* PhD thesis, Georgia Institute of Technology, August 1993.

[221] VANCE, P., BARNHART, C., JOHNSON, E. L., AND NEMHAUSER, G. L. Solving Binary Cutting Stock Problems by Column Generation and Branch-and-Bound. *Computational Optimization and Applications 3* (1994), 111–130.

[222] VASQUEZ-MARQUEZ, A. American Airlines Arrival Slot Allocation System (ASAS). *Interfaces 21*, 1 (January-February 1991), 42–61.

[223] WEDELIN, D. An Algorithm for Large-Scale 0-1 Integer Programming with Application to Airline Crew Scheduling. *Annals of Operations Research 57* (1995), 283–301.

[224] WELLS, A. *Air Transportation: A Management Perspective.* Wadsworth Publishing Company, 1994. Third Edition.

[225] WREN, A., SMITH M., AND MILLER, A. Complementary Approaches to Crew Scheduling. In *Computer Scheduling of Public Transport 2* (1985), J.-M. Rousseau, Ed., North Holland Publishing Company, pp. 263–278.