

AIRLINE PLANNING UNDER UNCERTAINTY

by
Shervin AhmadBeygi

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Industrial and Operations Engineering)
in The University of Michigan
2008

Doctoral Committee:

Assistant Professor Amy M. Cohn, Chair
Assistant Professor Damian Beil
Associate Professor Marina Epelman
Assistant Professor Amitabh Sinha

© Shervin AhmadBeygi 2008
All Rights Reserved

ACKNOWLEDGEMENTS

I wanted to thank all the people who helped me during different stages of my research in the graduate school. First, many thanks to my academic advisor, Professor Amy Cohn who gave me great opportunities to learn. I would like to greatly appreciate her encouraging role in many ways.

I wanted to also acknowledge the help of students who have been closely involved in this research. Particularly, many thanks to Mark Sytsma, Kristen Neubauer, Bobby Vaziri and Marshall Weir for their great help in implementing the ideas introduced in Chapter V. I wanted to also thank the technical team in Center for Advanced Computing for helping me in the process of implementing the parallel processing code for the crew pairing generator. Special thanks to Amadi Nwankpa for his invaluable help with CPLEX. Also thanks to Doctor Joe Norman for giving us access to his high performance computing platform.

I should also appreciate Yihan Guan for helping me formalize the propagation trees and define the corresponding measures and metrics introduced in Chapter III. Many thanks to Siti Mastura Daud who helped me validate the optimization models for re-allocating slack. Also special thanks to Marcial Lapp for his enormous help in developing and coding the simulation model discussed in Chapter IV. I also appreciate Professor Katta Murty for encouraging me to develop linear programming models of Chapter IV.

Many thanks to our friends from the airline industry who provided us with the

data and technical feedback. This research would not have happened without their great support. Also special thanks to Doctor Peter Belobaba from MIT for his pivotal role and brilliant ideas. Many thanks to Sloan Industry Studies for partially sponsoring this research. I also greatly appreciate the helpful comments I received from academic and industry practitioners during my conference presentations. In particular, I should thank Professors Mark Daskin, Thomas Vossen, Andrew Schaefer, Stephan Karisch and Goran and Mirela Stojković.

I also greatly appreciate the help and support I received from my dissertation committee, Professor Beil, Professor Sinha and Professor Epelman. I appreciate Professor Beil's encouraging comments as well as his help in my job search process. I am greatly thankful of Professor Sinha for his comments especially on the existence of directed acyclic graphs which can potentially sophisticate the delay propagation structure. I also appreciate Professor Epelman's encouraging ideas as well as her help in fixing several errors in the dissertation.

I appreciate Professor Larry Seiford for his important role throughout my graduate studies in Michigan. I should also thank him and Professor Steve Pollock for their encouraging comments in my job search process. I also acknowledge the great experience I had working with Professors Mark Van Oyen, Dushyant Sharma, Judy Jin, Romesh Saigal, David Kaufman and Omer Tsimhoni.

I am grateful of all the assistance I received from the IOE department's staff since I started as a PhD student in Michigan. Special thanks to Tina Blay and her family for their invaluable support. Also thanks to Celia Eidex Ciemnozolvsky, Nancy Murray, Mary Winter, Gwen Brown, Matt Irelan and Wanda Dobberstein for helping me with the administrative work. Many thanks to Chris Konrad for providing all kinds of hardware and software related help and Olof Henrik Minto

(Mint) and Rod Capps for their technical assistance.

I am also thankful of my great friends, Chia-Wei Kuo, Stan Dimitrov, Pierre-Yves Brunet, Damon Williams and Tara Terry, who shared their knowledge of LaTeX with me. Special thanks to Sarah Root and Shankara Kuppa for their kindness, encouragement and constructive criticism during my studies. I should also thank my colleagues Richard Chen, Mustafa Sir and Ada Barlatt for sharing their programming skills with me. Many thanks to my friends Soroush Saghafian, Kamran Payanbar, Esra Sisikoglu, Katrina Appell, Luz Caudillo, Eren Cetinkaya, Tim Lortz, Tim Maull, Joy Oguntebi, Murray Pyle, Minsuk Suh, Warren Sutton, Arleigh Waring, Zhibin Yang, Yu Zhou, Cindy Veenstra, Jonathan Helm and Jin-kyu Park who made my life as a graduate student a lot easier!

Finally, I should greatly acknowledge the support and encouragement of Hoda Parvin. Certainly, I could not have been successful, especially during my final year as a PhD student, without her great dedication. I am also grateful of my parents who had an inspiring role throughout my life. I owe all my achievements to them.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	ii
LIST OF FIGURES	vii
LIST OF TABLES	ix
ABSTRACT	xi
CHAPTER	
I. Introduction	1
II. Background and Literature Survey	6
2.1 Airline Planning Process	6
2.1.1 Schedule Generation	6
2.1.2 Fleet Assignment	7
2.1.3 Maintenance Routing	8
2.1.4 Crew Scheduling	8
2.2 New Challenges in the Airline Planning Process	8
2.2.1 Integrated Planning	10
2.2.2 Robust Planning	10
2.2.3 Recovery Models	11
III. Analyzing Airline Delay Propagation	12
3.1 Introduction	12
3.2 Analytical Framework	15
3.2.1 Motivation	15
3.2.2 Literature Survey	17
3.2.3 Propagation Trees	19
3.2.4 Study Parameters	23
3.3 Empirical Analysis	25
3.3.1 Case Study	25
3.3.2 Observations	43
3.4 Conclusions and Future Research	47
IV. Decreasing Airline Delay Propagation By Re-Allocating Scheduled Slack	49
4.1 Introduction	49
4.2 Literature Survey	51
4.2.1 Airline Planning Problems	52
4.2.2 Robust Planning	52

4.2.3	Flight Re-timing Models	53
4.3	Models for Re-Allocating Planned Slack	55
4.3.1	Main Idea	55
4.3.2	Single-Layer Model	58
4.3.3	Multi-Layer Model	62
4.3.4	Simultaneous Delays Model	67
4.4	Computational Experiments	70
4.4.1	Goal 1 – Tractability	72
4.4.2	Goal 2 – Impact	73
4.4.3	Goal 3 – Validity	76
4.4.4	Discussion	77
4.5	Conclusions and Future Research	80
V. An Integer Programming Approach to Generating Airline Crew Pairings		82
5.1	Introduction	82
5.2	The Crew Pairing Problem	84
5.2.1	Problem Statement	84
5.2.2	Formulation	86
5.2.3	Solution Techniques	88
5.3	An Integer Programming Approach to Generating Pairings	91
5.3.1	Motivation	91
5.3.2	Mathematical Model	92
5.3.3	Discussion	101
5.4	Initial Computational Experiments	109
5.4.1	Generating Initial Columns	109
5.4.2	143-Flight Network	110
5.4.3	329-Flight Network	112
5.5	Enhanced Computational Experiments	114
5.6	Conclusions and Future Research	116
VI. Conclusion and Future Research		118
BIBLIOGRAPHY		121

LIST OF FIGURES

Figure

1.1	Major U.S. airlines net profit (loss)[23].	1
1.2	The increasing trend in delayed flights [24].	2
1.3	Delay causes among all major US airports in Dec 2007 [24].	3
2.1	Airline planning sub-problems.	7
3.1	An example of a propagation tree.	19
3.2	Propagation trees corresponding to the maximum severity.	27
3.3	Propagation trees corresponding to the maximum depth.	29
3.4	Propagation trees corresponding to the maximum magnitude.	31
3.5	The effect of higher depth ratios on increasing magnitude.	32
3.6	The overall trend corresponding to the severity of the propagation trees.	36
3.7	The overall trend corresponding to the depth of the propagation trees.	36
3.8	The overall trend corresponding to the magnitude of the propagation trees.	37
3.9	Average total propagated minutes of the propagation trees.	38
3.10	Trends in the propagation measures categorized based on the departure time of the flights.	39
3.11	Trends in the propagation measures categorized based on the departure station size.	42
4.1	Visualizing constraint (2).	61
4.2	Visualizing <i>worst-case</i> propagation trees.	64
4.3	An example of simultaneous root delays where the surrogate objective function <i>overestimates</i> the delay propagations.	67
4.4	An example of simultaneous root delays where the surrogate objective function <i>underestimates</i> the delay propagations.	68

4.5	An example of a directed acyclic graph. In this case, flight f_3 has more than one parent.	79
5.1	The convergence trend in the master problem for 329 flight network.	113

LIST OF TABLES

Table

1.1	Direct costs of delays in the U.S. airline industry. Costs based on data reported by U.S. passenger and cargo airlines with annual revenues of at least \$100 million [23].	3
3.1	the sub-routine for constructing propagation trees	25
3.2	Severity breakdown for root delays of 180 minutes.	28
3.3	Depth breakdown for root delays of 180 minutes.	30
3.4	Depth ratio breakdown for root delay of 180 minute.	30
3.5	Magnitude breakdown for root delays of 180 minutes.	33
3.6	System-wide statistics on delay propagation: severity	34
3.7	System-wide statistics on delay propagation: depth	35
3.8	System-wide statistics on delay propagation: magnitude	35
3.9	System-wide statistics on delay propagation: total propagated delay minutes	38
4.1	Size of the instances and their corresponding run times.	73
4.2	Reduction in the expected delay propagation – comparing the original schedule schedules with SLM.	74
4.3	Reduction in the expected delay propagation – comparing the original schedule schedules with MLM.	74
4.4	Comparing the simulation results with the surrogate objective function value. . . .	76
4.5	Reduction in delay propagations (simulation results) – comparing different schedules.	76
4.6	95% confidence intervals on estimated reductions in delay propagation.	77
5.1	A counter example for dominance in MLSP	90
5.2	Generating optimal columns for 143 flight network.	111
5.3	Generating feasible columns for 143 flight network.	111
5.4	Generating feasible columns for 329 flight network.	112

5.5	Details of the computation for 329 flight network (time units are seconds).	112
5.6	Details of the computation for 329 flight network (time units are percents).	113
5.7	Average run times for 329 flight network.	113
5.8	Size of the IP-based sub-problems	114
5.9	Comparing the performance of the original and the enhanced IP for the 329-flight network.	115

ABSTRACT

In this dissertation, we study the impact of uncertainty associated with potential delays on the operational performance of the airline plans, considering different stages of the airline planning process. In the first part of this dissertation, we investigate the potential for flight delays to propagate throughout a passenger airline network. We also define metrics to quantify delay propagation by analyzing the connections in a flight network. As the computational results demonstrate, delays can sometimes propagate substantially and have a major impact on network efficiency. Therefore, in the second part, we develop a linear programming approach to strategically use slack in the network in order to mitigate the impact of disruption without increasing cost. In these models we allow flight departure times to change within a given time window while keeping the original connections feasible but re-allocating slack where it can best be utilized. The motivation for the third part of this dissertation stems from the fact that re-timing the schedule per se cannot fully capture all the opportunities for improved robustness. Instead, it may also be necessary to change the fleet assignment and crew schedule. Thus, in a robust planning tool, it is necessary to schedule flights across all fleet-types simultaneously. This, in turn, increases the number of flights to be considered which dramatically impacts tractability. Therefore, it is essential to generate crew pairings (sequences of flights that can be covered by a single crew) on larger flight networks. In the third part, we present a new, integer programming-based approach to supplement the existing techniques for generating crew pairings.

To demonstrate the performance of this approach, computational experiments based on data from a major U.S. carrier are presented.

CHAPTER I

Introduction

For the past decade, the US passenger airlines industry has gone through a severe economic crisis. Factors such as competition, increasing fuel costs in addition to delays and disruptions have resulted in a total of \$35 billion net loss over the period of 2001-2005 (on average \$7 billion per year, see Figure 1.1). Airlines have been trying to cover this loss partly by tightening their flight schedules which in turn has caused congestion of airports as well as the airspace. This, in turn, has made flight schedules more vulnerable to disruptions.

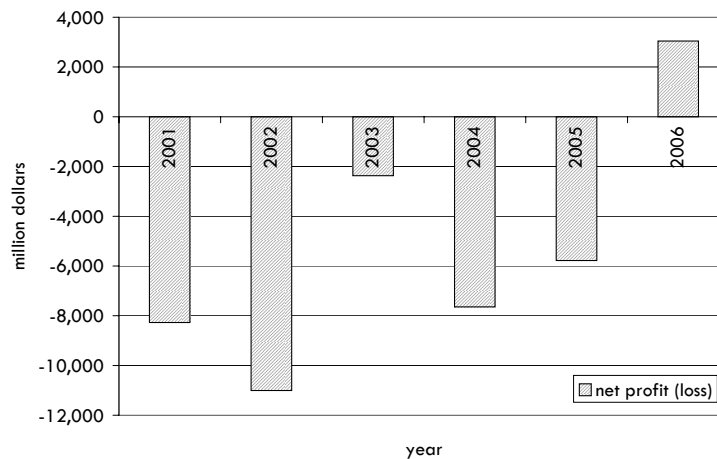


Figure 1.1: Major U.S. airlines net profit (loss)[23].

Consequently, airline delays have increased substantially over the past five years (see Figure 1.2). It has been estimated that in 2006, 116.5 million delay minutes (up

five percent from 2005) drove \$7.7 billion in direct operating costs (up 11 percent from 2005) for the major U.S. airlines. On average, extra fuel consumption and crew time are estimated to have cost \$42.55 per minute. A detailed breakdown of direct cost impacts of delays on major U.S. airlines is given in Table 1.1. Moreover, delays can also impose additional “indirect” or “invisible” costs to airlines such as lost productivity, accommodation for passengers with lost itineraries and potential lost profit due to the loss of good image. These additional costs are not easy to quantify but potentially have an even more destructive impact on airlines profit.

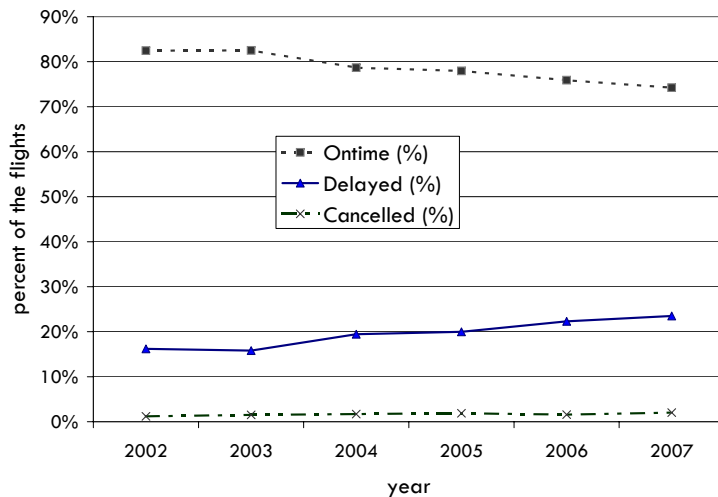


Figure 1.2: The increasing trend in delayed flights [24].

Clearly, some airline delays are due to *independent*, random events such as bad weather or mechanical failures. However, a considerable fraction (roughly one-third, see Figure 1.3) of total delay minutes are caused by *late arriving aircraft*. Delays of this nature are not independent. Instead they are a consequence of another delay in an upstream flight. More specifically, due to sharing of the same resources (aircraft or crews) an independently occurring delay can *propagate* to downstream flights.

The main objective of this dissertation is to analyze the impact of uncertainty caused by delays on different stages of the airline planning process and to develop

Direct Operating Costs (2006)	\$ Per Minute	Annual Delay Costs (\$ millions)
Fuel	28.31	3,296
Crew	14.25	1,659
Maintenance	10.97	1,277
Aircraft Ownership	9.18	1,069
Other	3.1	361
Total	65.80	7,663

Table 1.1: Direct costs of delays in the U.S. airline industry. Costs based on data reported by U.S. passenger and cargo airlines with annual revenues of at least \$100 million [23].

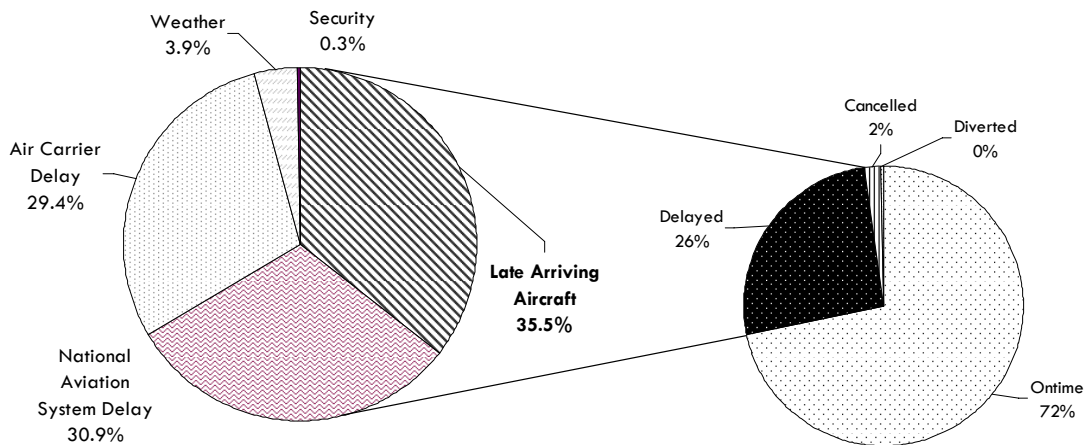


Figure 1.3: Delay causes among all major US airports in Dec 2007 [24].

procedures to reduce their disruptive impacts.

This dissertation is organized as follows: In Chapter II we discuss the airline planning process and briefly explain different sub-problems solved at each stage of this process. Then we present the new planning challenges facing the airline industry. In Chapter III we present an approach to investigate the potential for airline delays to propagate. We also discuss different metrics to evaluate delay propagation in airline plans. We then present the results of an empirical study performed on two major U.S. airlines. In Chapter IV we present a linear programming-based approach to mitigate the expected delay propagation by re-allocating the slack in a flight network. In the presented models we allow flight departure times to change marginally while keeping the original connections feasible. We discuss the effectiveness of this approach by presenting computational results using data from a major U.S. carrier. The motivation for the Chapter V of this dissertation stems from the fact that re-timing the schedule per se cannot fully capture all the opportunities for improved robustness. Instead, it may also be necessary to change the fleet assignment and crew schedule. Therefore in order to increase robustness in the airline plans it is often necessary to schedule flights across all fleet-types simultaneously. This, in turn, needs simultaneous consideration of fleet-types and results in larger flight sets. This increase in the number of flights to be considered has a dramatic impact on the size and tractability of corresponding crew pairing problems. In Chapter V, we present a new, integer programming-based approach to supplement the existing techniques for generating crew pairings. The final conclusion and future research are presented in Chapter VI.

Here are the main contributions of this dissertation:

Chapter III of this dissertation makes an important contribution towards under-

standing the relationship between planned airline schedules and their operational performance by introducing the idea of propagation trees as a visual and quantitative tool for assessing the ramifications of individual flight delays throughout the network. It also proposes several new metrics for quantifying the impact of delays as they propagate through the network which, in turn, are useful in conducting a quantitative analysis of planned airline schedules and lays the groundwork for future research on incorporating measures of potential delay propagation in the airline schedule planning process.

The primary contribution of Chapter IV is in developing models that can diminish delay propagation by re-timing the flight schedule. These models can take into account delay propagations caused by aircraft, crew members, connecting passengers, as well as other shared resources.

The contribution of Chapter V is in presenting a new way of modeling complex non-linear constraints and cost functions in a scheduling context. This approach also enables researchers to develop a crew pairing generator that can facilitate research in challenging research areas such as integrated airline planning, robust planning, and automated recovery.

CHAPTER II

Background and Literature Survey

2.1 Airline Planning Process

Airline operations are highly complicated processes, concerned with the control of many expensive, tightly-constrained, and interdependent resources such as crews, aircrafts, airports, and maintenance facilities. From very early on, the OR community has been actively involved in seeking ways to develop more efficient plans [3].

The airline planning process is classically decomposed into four sub-problems which are typically solved sequentially: schedule generation, fleet assignment, maintenance routing, and crew scheduling (see Figure 2.1). Here is a brief description of each of these sub-problems:

2.1.1 Schedule Generation

The objective of the *schedule generation* problem is to determine what markets (i.e., origin-destination itineraries), frequencies, and times to fly in a given period of time, taking into account both projected demand data and available resources. This problem has received the least attention from the OR community, largely because it involves so many factors that are hard to quantify, such as competition, market penetration and revenue management strategies. Nonetheless, some attempts have

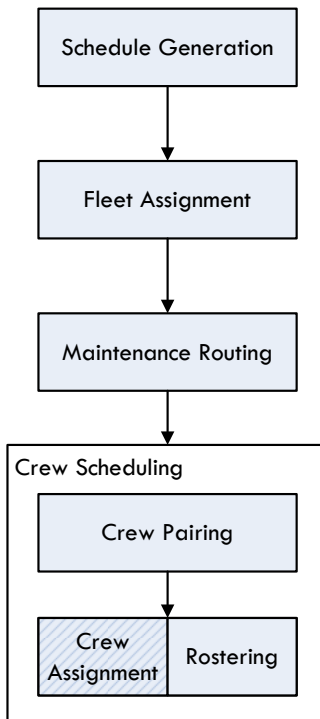


Figure 2.1: Airline planning sub-problems.

been made to model and solve this problem using optimization models. See Berge [9] and Teodorovic and Krcmar-Nozic [54] for a more detailed literature survey of these approaches.

2.1.2 Fleet Assignment

The *fleet assignment* problem determines which type of aircraft (for example, Boeing 757 or Airbus 300) should be assigned to each flight. The goal is to match demand and capacity as closely as possible, while still addressing the limitations imposed by the network structure. These limitations include the conservation of aircraft at each point in time and space, as well as limits on the number of aircrafts available in each fleet. Examples from the literature include Abara [2] and Rexing et al [43].

2.1.3 Maintenance Routing

The *maintenance routing* problem is primarily a feasibility problem which assigns specific aircraft to flights so as to ensure adequate opportunities for required maintenance checks. I.e, the assignment of individual aircraft to flights occurs in this stage. The objective of the maintenance routing problem is to find routings, or rotations, for each aircraft in a fleet. A rotation is a sequence of flights so that it starts and ends at the same location. The rotation assigned to each individual aircraft should meet certain maintenance stations at pre-planned intervals. Barnhart and Talluri [7] studied the solution methods for this problem in detail.

2.1.4 Crew Scheduling

The objective of the *crew scheduling* problem is to find the most cost-effective assignment of the crew (cockpit or cabin) to the flights. This is typically done in two major steps. The first step, the *crew pairing* problem, partitions the flights into *pairings* – strings of flights that start and end at a crewbase and can be flown by a single crew. The second step is the assignment of specific crews to each of these the pairings. This is known as either the *crew assignment* problem, in which crews bid for pairings (typical in the U.S.) or the *rostering* problem (common in Europe), which is posed as an optimization problem [31]. The crew pairing problem is studied in detail in Chapter V.

2.2 New Challenges in the Airline Planning Process

Two key obstacles have provided significant challenges in the development of tractable planning tools for the airline industry. The first is the sheer size and complexity of the networks. For example, a major U.S. airline company might have on the order of five hundred aircraft and offer over two thousand flights daily, in a flight

network of approximately 400,000 origin-destination pairs. Several hundred million passengers fly across such networks each year on U.S. domestic flights alone. The vast number of decisions to be made, and the complex interactions between different resources (crews, aircraft, airports, etc.), make the planning process a challenging task [23].

The second obstacle is the dynamic and stochastic nature of the system. Plans are typically developed under the assumption that the system will have the same characteristics every day of the planning horizon, which is often several weeks or months in length. In reality, there is significant variation in demand by day, and demand is a key driver of many planning decisions. Furthermore, even were a distinct plan to be built for each day (or were each day in the planning horizon assumed to have the same characteristics), many aspects of the system would still be subject to perturbations, both major and minor. For example, a flight's travel time depends on the day's wind speeds, its taxi time depends on congestion at the airport, maintenance issues can delay take-off, and so forth.

In order to deal with these obstacles, researchers have typically ignored the variability of the system, instead developing strictly deterministic models to ensure tractability. Furthermore, because airline planning models seek to maximize resource utilization, ignoring variability may lead to solutions with very little slack time. In such cases, the optimal plan may perform very poorly under disruptions because delays cannot be absorbed and instead propagate through the system.

In recent years, the already intense pressure on the airline industry has dramatically increased as a result of rising fuel costs, competitive pricing, increased congestion, and security concerns. It is therefore even more important to find high-quality plans that will perform well in practice. Improvements in both computer

speeds and OR algorithms [26] now make it possible to at least partially address these issues through *integrated planning*, *robust planning*, and the development of optimization-based *recovery tools*. Here is a brief review of these approaches:

2.2.1 Integrated Planning

Solving the airline planning sub-problems sequentially will not necessarily result in an optimal (or even feasible) solution to the overall problem. Therefore in *integrated planning* multiple sub-problems are simultaneously considered in an attempt to improve overall solution quality and feasibility while maintaining tractability. Examples of integrated planning literature include Cohn and Barnhart [13], Cordeau et al [14], and Klabjan et al [27], who partially integrate crew scheduling and aircraft routing, and Rexing et al [43] and Lohateponant and Barnhart [38], who permit some schedule generation decisions within the fleet assignment problem. Stojković et al [53] considered integrating the flight and pilot scheduling problems. A more detailed survey on integrated airline models can be found in Sandhu and Klabjan [48].

2.2.2 Robust Planning

Airline operations are subject to significant uncertainties. Disruptions often occur as a result of weather conditions, unplanned maintenance issues, safety checks, security concerns, and more. The goal of *robust planning* is to generate schedules that are less sensitive to disruptions. This is a demanding task, given both the size and the uncertainty of the networks. There have been two major types of approaches to robust airline planning in the literature. In the first approach, a stochastic programming model is used to capture uncertainties. For instance, Yen and Birge [59] developed a stochastic programming model for the airline crew scheduling problem and adopted a delay branching algorithm to solve the resulting problem. Alterna-

tively, Rosenberger et al [45] used a simulation model to implement a stochastic model for the crew scheduling problem and also to evaluate recovery policies. The second approach to modeling disruptions is to define surrogate problems that inherit the stochastic nature of the problem. For example, Rosenberger et al [44] showed that fleet assignments with less hub connectivity and more short cycles perform better operationally. Schaefer et al [49] modeled the crew scheduling problem under uncertainty using approximated expected pairing costs. Klabjan et al [28] defined a *regularity* measure as one way to capture robustness and considered maximizing this measure in addition to minimizing the total cost. A more detailed literature survey on robust airline planning is provided in Chapter IV.

2.2.3 Recovery Models

While robust planning models try to avoid disruptions proactively, *recovery models* seek the best way of reacting when disruptions do occur, so as to minimize their impact on the system and prevent propagation. Therefore, recovery models are often studied within the robust planning context [45]. Clarke et al [12] explained the role of an airline's operations control center in mitigating the impact of irregularities on operations. Yan and Yang [58] developed a framework to handle schedule perturbations caused by aircraft breakdowns. Abdelghany et al [2] developed a decision support tool to automate crew recovery during irregular operations. Lettovský et al [36] developed a real-time recovery plan to restore a disrupted crew schedule. Yu and Qi [60] have studied the recovery models used by United Airlines in the context of disruption management. More extensive surveys of the literature on recovery models can be found in Kohl et al [32] and Bratu and Barnhart [10].

CHAPTER III

Analyzing Airline Delay Propagation

3.1 Introduction

The operation of a passenger airline requires the allocation of resources and development of schedule plans over complex networks. A large airline can operate over a thousand flight departures per day with several hundred aircraft and thousands of cockpit and cabin crew employees. These resources are costly – the direct operating cost of a 183-seat Boeing 757 aircraft was over \$4300 per hour in 2006. The efficient utilization of costly resources is thus one of the key challenges faced by airlines hoping to control operating expenses in order to generate profits in an increasingly competitive fare environment [24].

The operations research community has played a significant role in developing airline planning tools with the aim of optimizing the utilization of these costly resources ([5], [7]). In these optimization tools, which typically assume deterministic flight times and other parameters, it is desirable to generate schedules that have little if any slack between flights – by *turning* crews and aircraft quickly, (i.e., having them connect from one flight to another with minimal time in between) greater utilization of these resources can be achieved and unit operating costs can be minimized.

In practice, however, flight times are *not* deterministic. Departure delays arise due

to mechanical problems, weather delays, ground holds, and other sources. Flights that depart on time can still be delayed in arrival due to causes such as air traffic control issues or re-routings to avoid inclement weather. In isolation, these delays are themselves costly. In a network structure, they can have an even greater impact – without adequate slack to absorb an initial *root delay*, subsequent flights may also be delayed as they await aircraft and crews from the initially delayed flight. We refer to this as *delay propagation*.

In fact, one would expect to see an inverse relationship between the planned level of resource utilization in a schedule and that schedule’s operational robustness. In a planned schedule with high resource utilization, there is limited slack. This in turn limits the opportunity to absorb flight delays, which must instead propagate to subsequent flights. But what is the nature of this relationship? How can it be incorporated in the planning process to produce “better” schedules? And what constitutes “better” – how should the deterministic costs of an airline plan be traded off against the potential and much more uncertain costs of delay? These are all important and challenging questions that are beginning to receive significant attention from both the airline industry and the academic community.

To assist in these efforts, we have undertaken an empirical study of passenger airline flight networks and their potential for delay propagation as a function of their planned schedule. This study is based on flight data from two U.S. carriers, one traditional “legacy” hub-and-spoke carrier and one “low-fare” carrier operating a predominantly point-to-point network. Using the idea of *propagation trees* as our foundation, we examine how any single given flight delay, in the absence of other flight delays, can propagate through the network (the structure is a tree because one flight uses multiple resources, such as cockpit crews and aircraft, and therefore each

flight delay can directly cause multiple subsequent delays, which in turn can continue to branch further). We analyze these trees for all flights in a given time period in order to gain insight into the distribution of slack throughout the system and the implications of this on delay propagation. We then use this analysis to address commonly-held assumptions about how delays propagate, provide insights into the relationship between planned resource utilization and operational robustness, and raise questions for further study.

This chapter of the dissertation makes several important contributions towards understanding the relationship between planned airline schedules and the operational performance of these schedules, a requisite precursor to developing more robust plans. First, we introduce the use of propagation trees as a visual and quantitative tool for assessing the ramifications of individual flight delays throughout the network. Second, we propose several new metrics for quantifying the impact of these initial delays as they propagate through the network. Third, we use these metrics to conduct a quantitative analysis of planned airline schedules, gaining insights into the relationship between scheduled slack and delay propagation. Fourth, we extend this analysis to substantiate (in some cases) and disprove (in other cases) commonly-held assumptions about delay propagation. Finally, we lay the groundwork for future research on incorporating measures of potential delay propagation in the airline schedule planning process.

The chapter is organized as follows. In Section 3.2, we outline the details of the study. We present our analysis in Section 3.3. Section 3.4 contains our conclusions and suggested areas for future research.

3.2 Analytical Framework

3.2.1 Motivation

Consider a (hypothetical) airline plan that maximizes resource utilization in the sense that, for every crew and for every aircraft, the time between two consecutive flight assignments is as short as possible. [Depending on the context, this time is referred to as *connection time*, *turn time*, *sit time*, or *ground time*.] There is of course some minimum time between any pair of flight assignments that must be observed – for example, an aircraft cannot be assigned to two subsequent flights unless there is adequate time between them for passengers from the first flight to de-plane, catering and cleaning tasks to be completed, and new passengers to board. We begin by assuming, in this hypothetical plan, that all crew and aircraft assignments exactly satisfy the minimum time between assignments. Such a planned schedule is ideal from the perspective that aircraft are being fully utilized and crews are not being paid for any excess time between flights.

Suppose further that this schedule is implemented, and that an arbitrary flight is delayed in departure by thirty minutes (for example, due to a mechanical problem that must be fixed before take-off). Assuming that this delay is not compensated for by increasing the travel speed, the flight will have a thirty minute arrival delay as well. Because the cockpit crew does not have any extra slack time before their next flight, this thirty minute delay will propagate to that flight as well, causing it to also be delayed in take-off by thirty minutes. If the crew and the aircraft do not stay together, then the aircraft's next flight will also experience a thirty-minute flight delay. The cabin crew could cause a third thirty-minute flight delay if they separate from the aircraft and cockpit crew. Likewise, if flights are held for connecting passengers from this flight, then these flights will be delayed as well.

Now consider this set of flights that have been delayed as a result of the initial flight delay. [We will refer to the initial flight delay as the *root delay*; it is also sometimes referred to as the *independent delay* in the literature (Lan et al [33]).] These flights will also arrive late at their destinations, resulting in a second layer of subsequent flight delays. In fact, a delayed aircraft will continue to propagate delay to all of its subsequent flights until the aircraft *goes off-rotation* (i.e., is removed from the flight network to undergo maintenance) or enters an overnight phase where there are no longer flights to be covered. Similarly, a crew (cockpit or cabin) will propagate delay to all of its subsequent flights until they *go off-duty* (i.e., their day’s schedule is completed). If the original crews and aircraft do not stay together for subsequent flights, then each of these resources will ultimately cause other resources (i.e., other crews and aircraft) to enter this stream of delays as well.

The situation we present here considers two improbable extremes – on one hand, a schedule without any slack at all and on the other hand a delay cycle that propagates indefinitely. In reality, other factors prevent schedules from fully utilizing all resources to their maximum levels (for example, the market demands that influence flight times and frequencies). Furthermore, recovery alternatives (canceling flights, calling in reserve crews, etc.) frequently prevent delays from propagating fully. Nonetheless, taking into account both objectives – maximizing the profitability of a schedule under ideal conditions; minimizing the propagation of delays in operation – presents an important challenge for airline planners.

It is also a difficult challenge, not only because the planning problems are themselves so complex and because the real-world environment is highly stochastic, but also because measures for quantifying robustness (and the value of this robustness) are not well-defined. In this study, we take an important first step in developing

metrics and tools for understanding passenger airline networks and the role of their structures in propagating flight delays.

3.2.2 Literature Survey

Within the literature, our research is most closely related to the work of Beatty et al [8], who also consider the impact of individual flight delays as they propagate across the network. Their research, which focuses on the relationship between time of day and delay propagation, simulates the movement of delayed cockpit crews, flight attendants, and aircraft through the network. In particular, they use the metric of “delay multiplier” to track the ratio of propagated delay minutes to the length of the initial delay.

More generally, several people have conducted empirical studies to better understand the causes and effects of flight delays. For example, Wang et al [56] use queuing models to examine how the response to propagated delays varies by airport. Queueing models are also used by Janic [25] to quantify the economic consequences of flight delays. Hsiao and Hansen [22] use a statistical model to investigate the impact of different factors such as time of day, congestion, and weather conditions on delay propagation. Tu et al [55] focus specifically on the long-term seasonal behavior of flight delays, while also taking into account additional short-term factors such as time of day. Rupp [46] considers factors such as weather condition, station type (hub vs. spoke), and seasonality to identify the most significant causes of delay.

Efforts to improve the robustness of airline schedules, so as to reduce the potential impacts of disruption, are also beginning to appear in the literature. Ehrgott and Ryan [16] focus on crew scheduling, considering the bi-criteria optimization problem of minimizing cost and maximizing robustness. Their measure of robustness considers the two options of keeping crew and aircraft together or increasing slack

between flights when a change of aircraft is scheduled, so as to minimize propagation. Shebalov and Klabjan [50] also consider a bi-criteria crew scheduling problem. In their case, the objectives are minimizing cost and maximizing the number of *move-up crews* (the number of flights for which there exists an alternate crew scheduled for a later departure that could be “moved up” to cover this flight if its scheduled crew is delayed). Schaefer et al [49] instead use an approximation of the *expected operational cost* of a crew pairing in solving the crew scheduling problem as a single-objective optimization problem. Yen and Birge [59] formulate the crew scheduling problem as a stochastic integer program in order to incorporate the impact of delays.

In Rosenberger et al [44], the focus is on incorporating robustness in the fleet assignment problem. By constructing fleet assignments and aircraft rotations with many *short cycles* that could be canceled without violating flow balance, they seek to provide greater opportunity for canceling a limited number of flights without significant network impact. Smith and Johnson [51] also seek to improve robustness in the fleet assignment by imposing *station purity*, a limitation on the number of fleet types that land at/depart from each airport, which in turn provides greater flexibility in recovery operations.

Finally, the work of Lan et al [33] presents an integrated model that simultaneously assigns fleet types and flight departure times, with an emphasis on minimizing the impact of delays on passengers.

We extend this literature by conducting a detailed investigation of the relationship between scheduled plans and the potential for delays to propagate in operations, a requisite precursor to developing tools for building more robust plans.

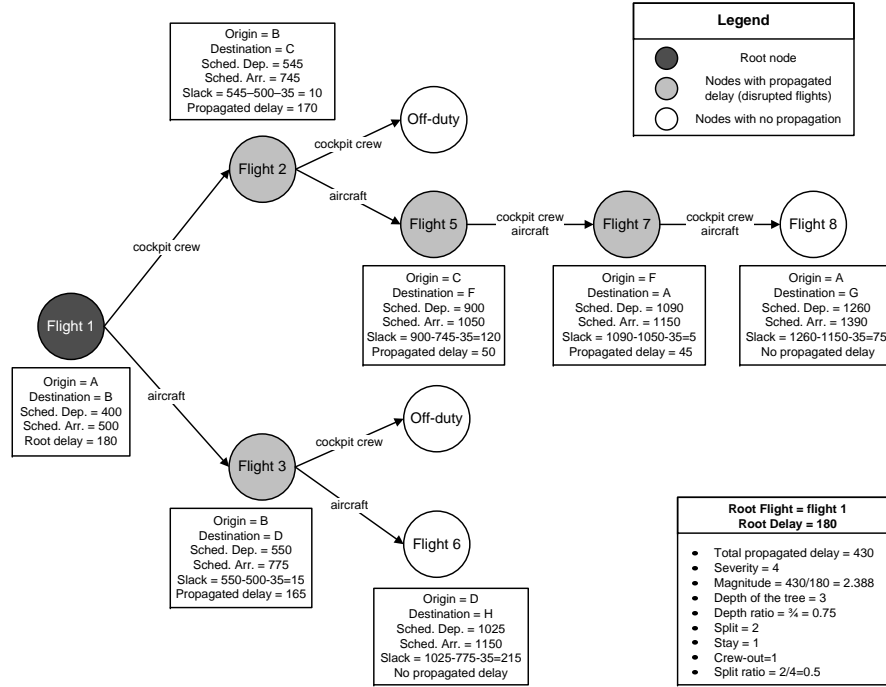


Figure 3.1: An example of a propagation tree.

3.2.3 Propagation Trees

A *propagation tree* is a simple but powerful visual and quantitative device that enables us to better understand how a root delay can, in the absence of other disruptions or schedule modifications, propagate through the network. Figure 3.1 provides an example of such a tree.

In this example Flight 1 is the root flight, which we suppose is delayed by 180 minutes. Note that this is an “independent” delay, caused by a mechanical problem, weather conditions, etc rather than because of an upstream flight delay. After landing, the cockpit crew of Flight 1 is scheduled to fly Flight 2 and the aircraft is scheduled to fly Flight 3. This is shown in Figure 3.1 by two arcs coming out of Flight 1. Assuming a minimum of 35 minutes between any two consecutive flight assignments, the cockpit crew has ten minutes of slack between Flights 1 and 2 in

the original schedule. Thus, if Flight 1 is 180 minutes late in arriving, then 170 minutes of delay will propagate via the cockpit crew to Flight 2. The cockpit crew from Flights 1 and 2 goes off duty after Flight 2 and thus does not propagate further. However, the aircraft used on Flight 2 continues on to Flight 5. Because this connection initially had 50 minutes of slack and now the aircraft is 170 minutes late in arriving, 120 minutes of delay will propagate to Flight 5. Given that both the crew and the aircraft of Flight 5 connect to Flight 7, only one downstream flight can be affected. In this case, because the connection had 5 minutes of slack in the original schedule, 45 minutes of delay will propagate. Finally, the crew and the aircraft again stay together to connect from Flight 7 to Flight 8. Because there is 75 minutes of slack in the schedule for this connection, the remaining 45 minutes of delay will be absorbed and there will be no further propagation of delay from this flight.

Similarly, the aircraft assigned to Flight 1 next connects to Flight 3. Because there is 15 minutes of slack scheduled in this connection, 165 minutes of delay will propagate to Flight 3. The crew from Flight 3 next goes off-duty and therefore does not propagate any further delay. The aircraft continues on, connecting to Flight 6. This connection has sufficient slack (215 minutes), however, and thus the remaining delay from this branch of the tree is also absorbed and there is no further propagation.

By examining this propagation tree in its entirety, we observe that the original delay of 180 minutes to Flight 1 also leads to an additional 430 minutes of downstream delay, collectively impacting four other flights.

Note that we have only considered the impact of delay on two resources – the aircraft and the cockpit crews. We do this for a number of reasons. The first is data availability – these are the two resources reported in the data provided by the supporting carriers in our preliminary study. The second is importance – these

are the two most costly resources, with cabin crews and passenger delays having less significant impact. The third is complexity – we felt that initially limiting our study to only two resources would enable us to develop an understanding of some of the critical drivers of delay propagation without being overwhelmed by the network interactions. We hope that the results of this first study will facilitate a second study that incorporates additional network resources (particularly, cabin crews and connecting passengers) as well.

We construct a distinct propagation tree for each scheduled flight and for a range of initial lengths of delay, to help develop an understanding of the relationship between the schedule and the potential for delays to propagate. But how can we process this collection of trees and their corresponding data? What information is of relevance?

We have developed several metrics that we suggest are of value in understanding how delays propagate:

- **Total propagated delay:** The sum of the delays (in minutes) imposed on downstream flights by an initial root delay in a propagation tree; note that the root delay is not included in this total. In Figure 3.1, the total propagated delay is 430 minutes.
- **Magnitude:** The ratio of total propagated delay to root delay. In Figure 3.1, the magnitude is 2.388. [This measure has also been referred to as *delay multiplier* (Beatty et al [8]).]
- **Severity:** The total number of disrupted flights, excluding the root flight itself. In Figure 3.1, the severity is 4.
- **Depth:** The number of nodes in the longest path in a propagation tree (not counting the root delay). In Figure 3.1, the depth is 3.

- **Depth ratio:** The ratio of depth to severity in a propagation tree. In Figure 3.1, the depth ratio is 0.75.
- **Stay:** The total number of nodes (disrupted flights) in which both the crew and the aircraft are the same as in the preceding node. In Figure 3.1, the stay is 1 (Flight 7).
- **Crew-out:** The total number of nodes (disrupted flights) in which the crew is not the same as the preceding node, because the crew in the preceding node has ended their pairing. In Figure 3.1, the crew-out is 1 (Flight 5).
- **Split:** The total number of nodes (disrupted flights) in which either the crew or the aircraft is not the same as the preceding flight, because these resources split to serve two different subsequent flights. In Figure 3.1, the split is 2 (Flights 2 and 3).
- **Split ratio:** The ratio of split to severity in a propagation tree. In Figure 3.1, the split ratio is 0.5.

In the remainder of this chapter, we limit our focus primarily to the first five metrics for the sake of brevity. We introduce the remaining four metrics as well, however, because they are valuable in further understanding the impact of keeping crews and aircraft together in the schedule.

Through the use of these metrics, we are able to quantitatively evaluate the relationship between an individual root delay and the rest of the network. This provides insights into the relationship between airline schedules and their robustness. In particular, we use these metrics to address “conventional wisdom” from the airline and academic communities about these relationships. For example, we consider the following commonly-held beliefs:

- Propagated delays create significantly more impact than the original root delays themselves.
- A single delay can “snowball” through the entire network, affecting a large number of subsequent flights.
- Scheduling aircraft and crews to stay together can help to mitigate the impact of disruption.
- Delays that occur early in the day cause greater propagation than delays later in the day.
- It is most important to prevent delay propagation early in the day (in other words, slack should be more pronounced in the early parts of the schedule).

We will revisit these claims in Section 3.3.2.

3.2.4 Study Parameters

In this study, we conduct analysis on data sets provided by two very different U.S. passenger airlines. The first is a traditional hub-and-spoke carrier that provides virtually full coverage of the U.S.; their daily flight schedule for the time period that we consider contains approximately 1700 flights. The second carrier is a point-to-point, “low-fare” carrier that focuses primarily on a specific and limited set of markets, many of them targeting leisure passengers. This carrier offers approximately 400 daily flights in the schedule that we consider.

For both carriers, we look at a single one-day “snapshot” of the schedule. Specifically, the carriers provided us with their complete schedule for a given day. This snapshot contains the complete set of domestic flights (international flights were not considered) scheduled to be operated on that day. For each flight, we were given the

origin and destination, scheduled departure and scheduled arrival times, scheduled *tail number* (i.e., unique identifier for a specific airplane), and scheduled cockpit crew. This information allows us to in turn identify all pairs of sequential flights sharing a common aircraft, a common crew, or both.

For *each* flight in these networks, and for several different potential root delay lengths, we construct a distinct propagation tree. Recall that a propagation tree looks at a single root delay in isolation, assuming no other concurrent delays in the network. It also assumes that all delays propagate until they are absorbed – we do not consider recovery options such as canceling flights or calling in backup (*reserve*) crews. In constructing these trees, we assume a minimum of 35 minutes between all pairs of sequential flight assignments. We also assume a constant *minimum rest period* of 9.5 hours for all crews between one day’s duty and the next. In addition to computing the individual metrics (as defined in Section 3.2.1) for each of these propagation trees, we also compile aggregate statistics, looking at flights grouped by origin and by time of day. The sub-routine we use in order to generate the propagation trees is shown in Table 3.1.

We analyze these results and discuss their implications in Section 3.3. Before doing so, we conclude this section by noting limitations of our study. First, we only take into account aircraft and cockpit crews. Other resources (in particular, cabin crews and connecting passengers) can cause additional delay propagation. Second, we do not take into account interactions between delays. In reality, there is often correlation between delays (in particular, due to weather conditions) and thus propagation trees will impact one another. Third, we do not consider recovery options (canceling flights, calling in reserve crews, etc.). One of the challenges of doing so is that these decisions are rarely codified by the airlines, but instead are typically made in

```

for each value of root delay
for each flight
{
  initialize the propagation tree
  initialize the list of nodes
  create the root node and add it to the list
  while there is a node in the list
  {
    calculate the slack between the flight in the current node and the succeeding flights
    if there is not enough slack
    {
      create a new node and add it to the list
      update the propagation tree statistics
    }
    delete the current node
  }
}
}

```

Table 3.1: the sub-routine for constructing propagation trees

an ad hoc manner by System Operations Control (SOC) personnel, based on their experience and intuition. Fourth, we do not weight the probability of root delays. In our aggregate data, all root delays are treated equally. Finally, our data sets are restricted to only two carriers, and one specific day in each carrier’s schedule. As a result, the analysis is by no means intended to make generalizable conclusions but rather to gain preliminary insights and to develop metrics and tools for further analysis.

3.3 Empirical Analysis

3.3.1 Case Study

In our case study, we consider flight data from two carriers, one hub-and-spoke legacy carrier and one point-to-point low-fare carrier. We look at all scheduled flights for a single day. The first network has approximately 1700 flights, the other approximately 400 flights.

For each of these flights, and for each potential delay, ranging from 15 minutes

to 180 minutes in increments of 15, we construct the propagation tree. [We do not consider delays longer than 180 minutes because such delays would typically lead to cancelation or other recovery methods, rather than delaying subsequent flights.] In our analysis, we examine both individual trees and the aggregation of their metrics.

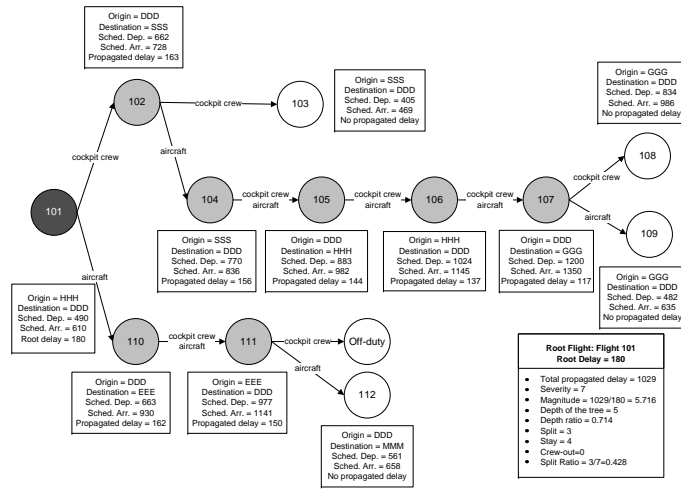
Highest-Impact Root Delays

We begin our analysis by first seeking to identify the maximum impact that any one individual root delay can have on the planned schedule. Specifically, we identify the maximum severity, depth, and magnitude that can be achieved as a result of a single root delay. [Of course, these levels will be achieved under the maximum root delay length, which is 180 minutes in our study.]

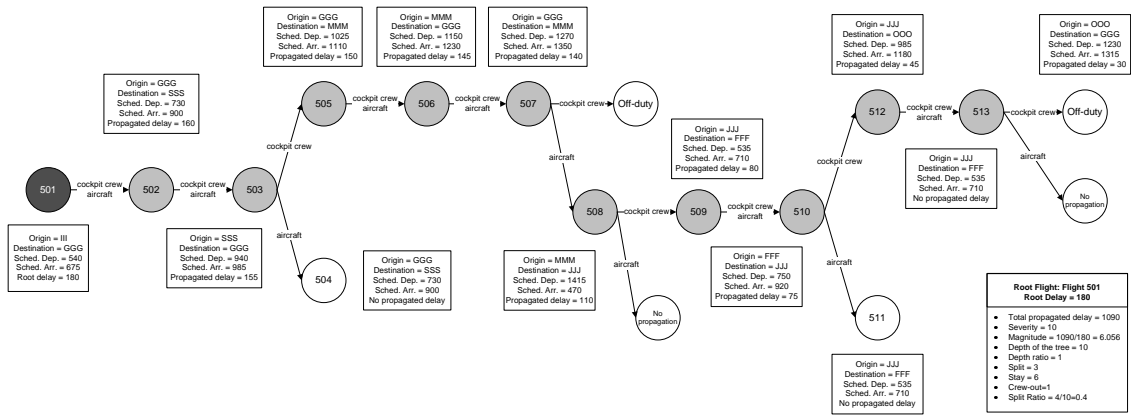
Maximum Severity Recall that severity refers to the *number* of downstream flights that are impacted as a result of a root flight delay. In the hub-and-spoke network that we consider, in the worst case a single flight delay can impact seven other flights (maximum severity). This occurs only in four cases, i.e., there are only four flights for which a 180 minute root delay can lead to seven other subsequent flight delays. The corresponding propagation tree for one of these cases is illustrated in Figure 3.2(a). Observe that this root delay leads to 1029 total propagated delay minutes, and thus a magnitude of 5.716. The depth of this tree is five.

We observe similar results in the point-to-point network (Figure 3.2(b)). In this case, there is a single flight that, in response to a 180 minute delay, impacts 10 other flights (maximum severity); the depth is also ten. This tree has a magnitude of 6.056, associated with 1090 total propagated delay minutes.

There are two interesting observations that stem from these results. First, it seems logical that those root delays leading to the highest severity would come as



(a) Hub-and-Spoke Carrier



(b) point-to-point Carrier

Figure 3.2: Propagation trees corresponding to the maximum severity.

a function of the most frequent splitting of resources – for example, each delayed flight in the propagation tree leading to two subsequent flight delays (one due to the aircraft and a second due to the crew), which in turn each lead to two subsequent delays, etc. However, the propagation trees that exhibit the greatest severity do not in fact show this sort of exponential growth. Instead, these trees have only one or two branches.

severity	hub-and-spoke		point-to-point	
	number of flights	percent of flights	number of flights	percent of flights
10	0	0.00%	1	0.24%
9	0	0.00%	1	0.24%
8	0	0.00%	3	0.73%
7	4	0.23%	4	0.98%
6	6	0.35%	5	1.22%
5	20	1.16%	14	3.41%
4	68	3.96%	18	4.39%
3	201	11.69%	36	8.78%
2	303	17.63%	65	15.85%
1	460	26.76%	99	24.15%
0	657	38.22%	164	40.00%
sum	1719	100.00%	410	100.00%

Table 3.2: Severity breakdown for root delays of 180 minutes.

The second observation is that these extreme severity cases are quite rare. Although the maximum severity levels are 7 and 10, respectively, for the two networks, the vast majority of flights have severities that are significantly smaller. Table 3.2 provides the breakdown of severity values for the complete set of flights. Specifically, for each severity value from zero (no delay propagation) to ten (the maximum severity), this table provides the number and percentage of flights achieving that severity given a 180 minute root delay. It is interesting to note that, even with a root delay as large as 180 minutes, more than 38% (40%) of the root flights do not propagate at all, and almost 98% (93%) of the root flights propagate to four or fewer downstream flights.

Maximum Depth “Depth” refers to the longest path in a propagation tree. In the hub-and-spoke network, the maximum depth of any propagation tree is six; this is achieved by two flights, one of which is depicted in Figure 3. In the point-to-point network, the maximum depth is ten. This is associated with the same flight (Figure 3.2(b)) that leads to the propagation tree of maximum severity for that carrier. Table 3.3 demonstrates that these extreme cases are again quite rare, and that the typical propagation tree has much lower depth.

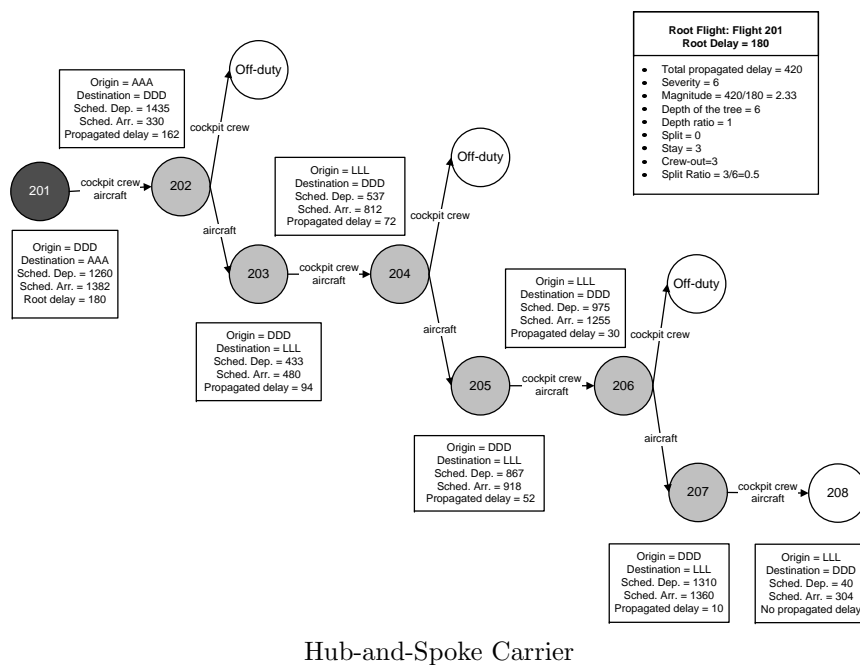


Figure 3.3: Propagation trees corresponding to the maximum depth.

It is interesting to note that there is not a dramatic difference between depth and severity – that is, a large number of the propagation trees are really *propagation chains*. In some cases, this is because the crew and the aircraft are scheduled to remain together. In those cases where the resources do separate, one resource may have enough to slack to absorb the disruption while the other propagates the delay. Alternatively, the crew may go off duty or the aircraft may go out of rotation and

thus not propagate the delay. Finally, we note that because our data sets do not include international flights, some resources terminate prematurely in our analysis.

depth	hub-and-spoke		point-to-point	
	number of flights	percent of flights	number of flights	percent of flights
10	0	0.00%	1	0.24%
9	0	0.00%	1	0.24%
8	0	0.00%	2	0.49%
7	4	0.23%	3	0.73%
6	2	0.12%	4	0.98%
5	20	1.16%	13	3.17%
4	68	3.96%	19	4.63%
3	202	11.75%	37	9.02%
2	302	17.57%	64	15.61%
1	468	27.23%	102	24.88%
0	657	38.22%	164	40.00%
sum	1719	100.00%	410	100.00%

Table 3.3: Depth breakdown for root delays of 180 minutes.

We further investigate the relationship between depth and severity in Table 3.4. Although it is commonly assumed that a major source of delay propagation in airline networks is the splitting of resources, we observe in this table that for many trees, such splitting does not occur. In fact, in fewer than 3% of the root delays is there a depth ratio strictly between zero (i.e., no propagation at all) and one (severity equals depth and hence no splitting).

depth ratio	hub-and-spoke		point-to-point	
	number of flights	percent of flights	number of flights	percent of flights
1	1033	60.09%	235	57.32%
(0,1)	29	1.69%	11	2.68%
0	657	38.22%	164	40.00%
sum	1719	100.00%	410	100.00%

Table 3.4: Depth ratio breakdown for root delay of 180 minute.

Maximum Magnitude We next consider the maximum magnitude, i.e., the ratio of propagated delay minutes to root delay minutes. In the point-to-point network, the maximum magnitude is 6.16, which corresponds to 1110 additional minutes of propagated delay as a result of the original root delay. In this tree (Figure 3.4(b)),

both the severity and depth are 9 (close to the maximum value for both of these metrics as well).

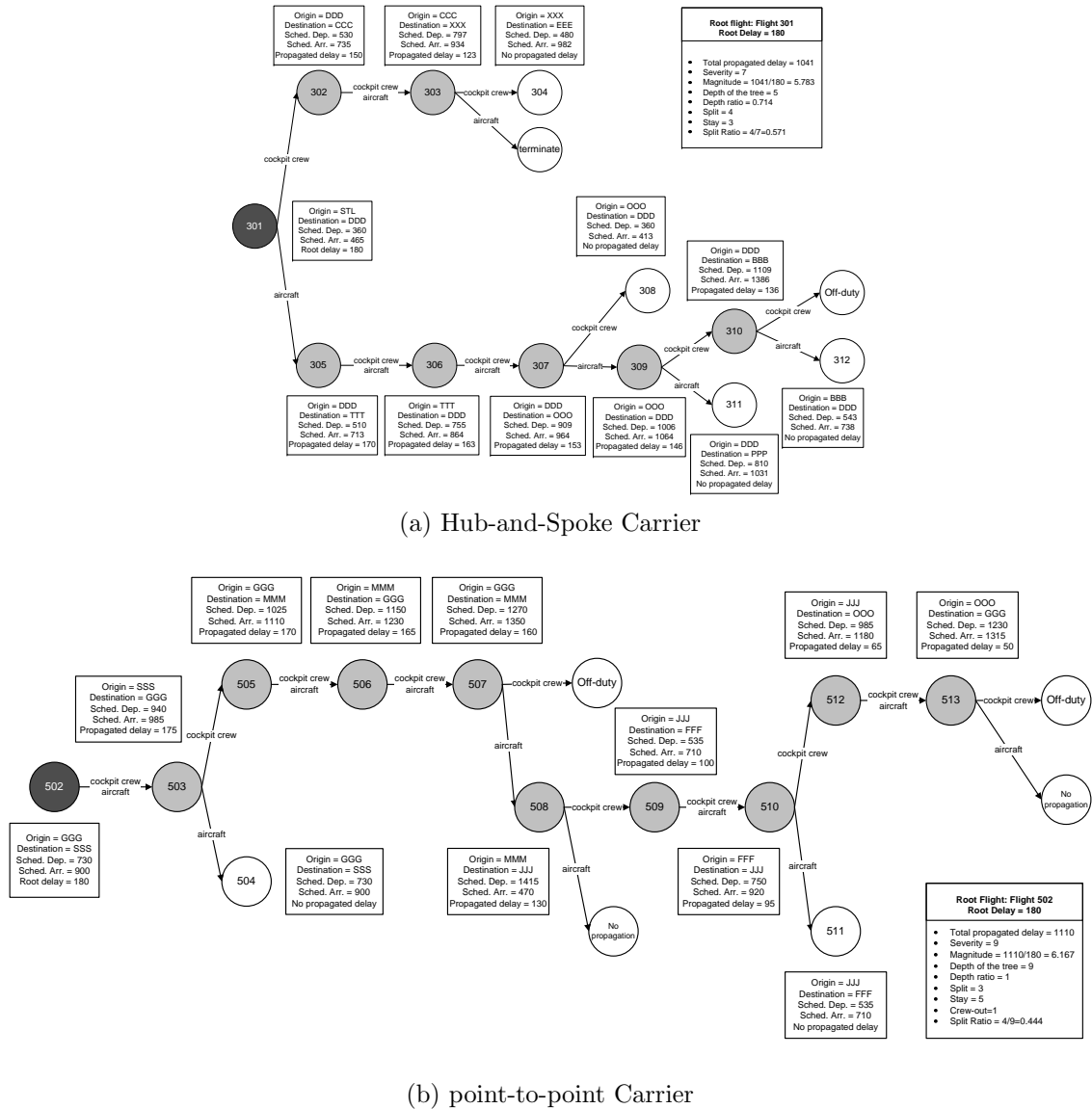


Figure 3.4: Propagation trees corresponding to the maximum magnitude.

In the hub-and-spoke network, however, the propagation tree with highest magnitude – 5.78, corresponding to 1041 additional minutes of propagated delay – looks quite different (see Figure 3.4(a)). In this case, the tree does in fact split, and has a depth ratio of about 0.70. In general, one would expect higher magnitude to occur

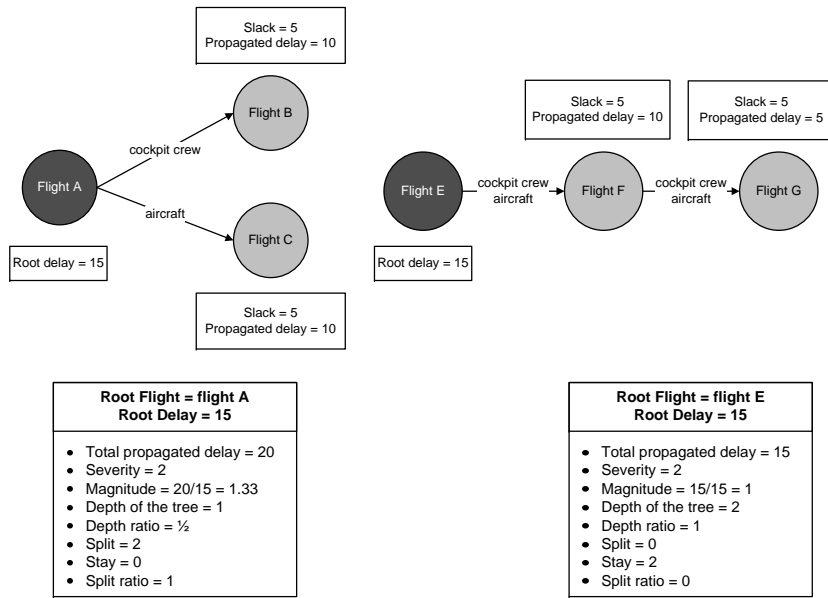


Figure 3.5: The effect of higher depth ratios on increasing magnitude.

in trees with more splitting. At each new level of the tree, the propagated delay is dampened by any slack between the connecting flights, and thus propagation trees with lower depth-to-severity ratios (i.e., more splitting) will tend to have higher magnitude.

This idea is demonstrated more clearly in Figure 3.5. In this figure, we have two different flights with root delays of fifteen minutes each. Both root delays lead to a severity of two, and all flights have a slack of five minutes. However, Flight A splits (depth equals one-half severity) whereas the crew and aircraft of flight E remain together for the next two flights (depth equals severity). As a result, Flight A yields 33% more propagated delay. More generally, assuming equal severity, we would expect to see greater magnitude values for propagation trees with lower depth ratios.

We conclude this section by providing a summary of magnitude values across the complete set of flights in Table 3.5.

magnitude	hub-and-spoke		point-to-point	
	number of flights	percent of flights	number of flights	percent of flights
(6,7]	0	0.00%	2	0.49%
(5,6]	3	0.17%	3	0.73%
(4,5]	12	0.70%	9	2.20%
(3,4]	62	3.61%	14	3.41%
(2,3]	198	11.52%	42	10.24%
(1,2]	316	18.38%	73	17.80%
(0,1]	471	27.40%	103	25.12%
0	657	38.22%	164	40.00%
sum	1719	100.00%	410	100.00%

Table 3.5: Magnitude breakdown for root delays of 180 minutes.

Range of Propagation Tree Characteristics

In the previous section, we identified the most extreme impacts of root delays, in terms of magnitude, depth, and severity. We also observed that these extreme cases were quite rare. Furthermore, they all occurred as a result of the most lengthy root delay (180 minutes). In this section, we present the complete set of root delays, including not only the full set of flights but also the full range (15 minutes to 180 minutes) of potential lengths of delay.

In Table 3.6 and Figures 3.6(a) and 3.6(b), we present system-wide statistics on propagation severity. Table 3.6 lists the range of root delay lengths (from 15 minutes to 180 minutes), the maximum severity achieved by any flight for that length of root delay, and the average severity across all flights given that length of root delay.

Observe that the maximum severity level quickly jumps in the initial increases in flight delay, but then remains constant (for the hub-and-spoke carrier) or grows much more slowly (for the point-to-point carrier) as the amount of delay increases. The average severity grows a little more steadily but nonetheless also quickly reaches near-maximum levels, then shows very little growth beyond this point. This fact is even more evident in Figures 3.6(a) and 3.6(b), which show the percentage of flights achieving each given severity level for each value of the root delay. Again, we

see a sharp change in the graph at a relatively low length of root delay, and then the graph remains nearly constant beyond this point. [Tables 3.7 and 3.8, and the corresponding Figure 3.7(a), 3.7(b), 3.8(a), and 3.8(b) demonstrate similar behavior for depth and magnitude as well.]

In other words, the network appears to reach a *saturation point*, and increasing the length of the root delay beyond this point does not dramatically change the corresponding severity. Why is this? Consider, for a given root flight, constructing the *exhaustive connection tree* – this is like a propagation tree, but at each flight in the tree, one or two new arcs are automatically created to represent the next flight (or flights) of the crew and aircraft (there is no notion of delay in this tree). Thus, the tree continues to grow until the resources leave the network (crews going off duty, aircraft going out of rotation), or the connection time exceeds 180 minutes and thus no delay would ever propagate.

root delay	hub-and-spoke			point-to-point		
	max	average (all)	average (only nonzero)	max	average (all)	average (only nonzero)
15	1	0.12	1.00	1	0.19	1.00
30	4	0.73	1.41	4	0.54	1.37
45	6	0.98	1.70	5	0.78	1.51
60	6	1.08	1.82	5	0.95	1.74
75	7	1.13	1.88	5	1.06	1.91
90	7	1.15	1.91	6	1.16	2.03
105	7	1.17	1.92	7	1.22	2.11
120	7	1.18	1.93	8	1.29	2.21
135	7	1.19	1.94	8	1.32	2.26
150	7	1.21	1.96	9	1.36	2.31
165	7	1.21	1.97	10	1.40	2.34
180	7	1.22	1.98	10	1.41	2.36

Table 3.6: System-wide statistics on delay propagation: severity

This exhaustive connection tree is a superset of all possible propagation trees for that root flight (i.e., the tree for each root length of delay). Different propagation trees from the same exhaustive connection tree can vary in two ways. First, they can

root delay	hub-and-spoke			point-to-point		
	max	average (all)	average (only nonzero)	max	average (all)	average (only nonzero)
15	1	0.12	1.00	1	0.19	1.00
30	3	0.72	1.40	4	0.54	1.37
45	4	0.97	1.68	5	0.78	1.51
60	5	1.07	1.81	5	0.95	1.74
75	5	1.11	1.86	5	1.05	1.89
90	5	1.14	1.88	6	1.15	2.01
105	5	1.15	1.89	7	1.20	2.08
120	5	1.16	1.90	8	1.26	2.16
135	5	1.17	1.91	8	1.29	2.21
150	5	1.18	1.92	9	1.32	2.25
165	6	1.19	1.93	10	1.36	2.28
180	6	1.20	1.94	10	1.37	2.28

Table 3.7: System-wide statistics on delay propagation: depth

root delay	hub-and-spoke			point-to-point		
	max	average (all)	average (only nonzero)	max	average (all)	average (only nonzero)
15	1.00	0.15	0.53	1.00	0.17	0.77
30	2.10	0.41	0.78	2.50	0.34	0.85
45	3.29	0.59	1.02	3.33	0.49	0.95
60	3.97	0.71	1.19	3.75	0.60	1.11
75	4.37	0.79	1.31	4.00	0.69	1.25
90	4.64	0.85	1.40	4.17	0.77	1.35
105	4.91	0.89	1.47	4.48	0.83	1.44
120	5.18	0.93	1.52	4.83	0.89	1.53
135	5.38	0.96	1.56	5.22	0.94	1.60
150	5.54	0.98	1.60	5.60	0.98	1.67
165	5.67	1.00	1.63	5.91	1.02	1.71
180	5.78	1.02	1.66	6.17	1.05	1.75

Table 3.8: System-wide statistics on delay propagation: magnitude

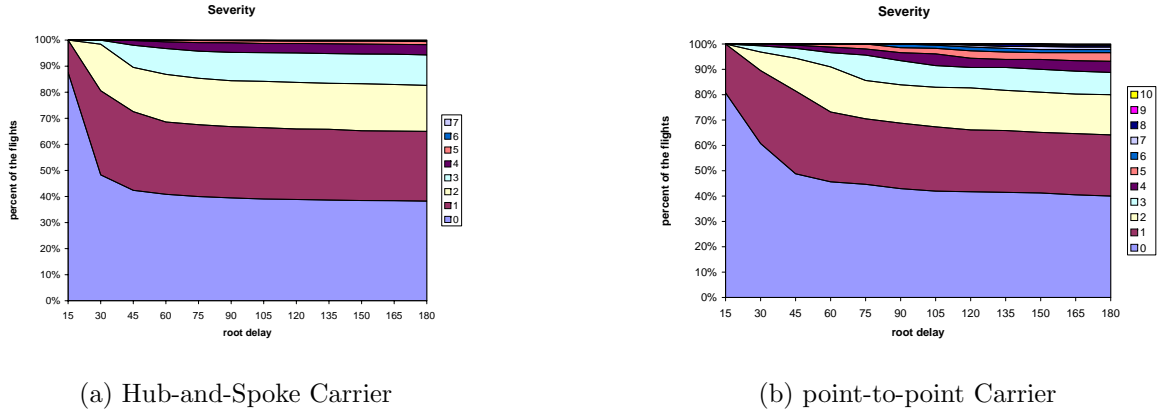


Figure 3.6: The overall trend corresponding to the severity of the propagation trees.

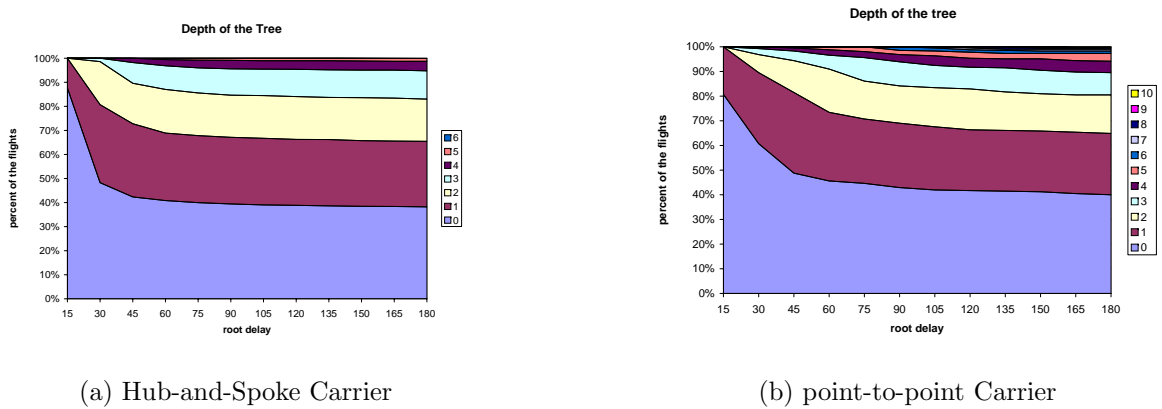


Figure 3.7: The overall trend corresponding to the depth of the propagation trees.

have different nodes (i.e., flights). Depending on the length of the root delay, one propagation tree may contain fewer nodes than another because the shorter delay is absorbed in one tree but the longer delay continues to propagate in another. A second way that propagation trees for the same root flight can vary is in the lengths of the arcs – that is the actual departure times of the delayed flights. If there is 30 minutes of slack between two connecting flights and the first flight experiences a 60 minute root delay, then 30 minutes will propagate across this connection, whereas in a second propagation tree corresponding to a 90 minute root delay, 60 minutes will instead propagate.

With this in mind, it is clear that at some point, as the length of root delay

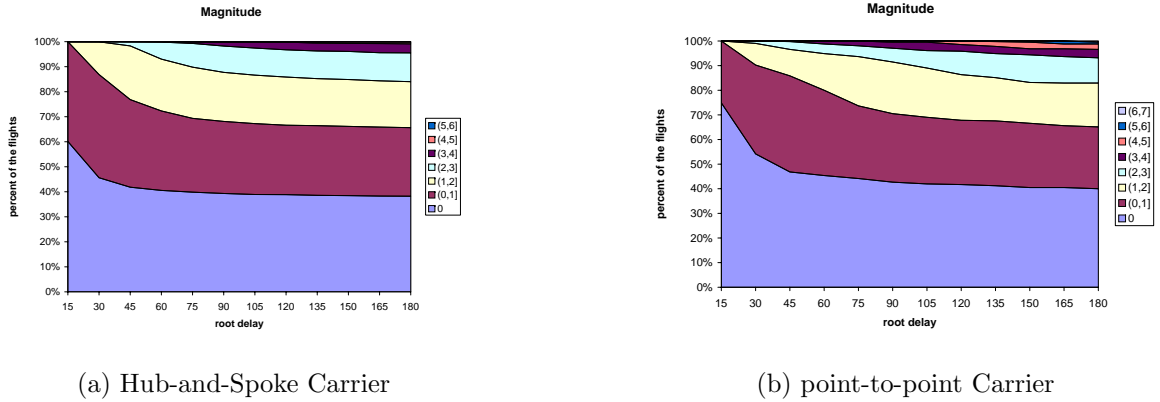


Figure 3.8: The overall trend corresponding to the magnitude of the propagation trees.

increases, the resulting propagation tree will eventually contain all of the nodes of the exhaustive connection tree. This in turn defines the severity and depth. As the length of root delay grows even longer, neither the severity nor the depth can increase – all flights that could be delayed are in fact being delayed. The total accumulated delay, however, does in fact continue to grow – this can be thought of as “stretching” the tree, with every flight (i.e., node) moving further forward in time. This is observed in Table 3.9 and Figures 3.9(a) and 3.9(b), which demonstrate that the total number of propagated delay minutes grows much more sharply across the shorter lengths of root delay, then becomes roughly linear (but certainly continues to increase) at this saturation point.

Range of Characteristics by Flight Category

In the previous section, we observed that many root delays do *not* propagate; those that do range significantly in the impact of their propagation. In this section, we question whether there are groups of flights that exhibit common behaviors in their propagation patterns. More specifically, we consider two factors: departure time of the flights (time of day) and origin station.

root delay	hub-and-spoke			point-to-point		
	max	average (all)	average (only nonzero)	max	average (all)	average (only nonzero)
15	15	2.24	7.93	15	2.52	11.58
30	63	12.25	23.51	75	10.33	25.47
45	148	26.44	45.83	150	21.91	42.60
60	238	42.43	71.65	225	36.11	66.37
75	328	59.20	98.60	300	51.91	93.71
90	418	76.45	126.24	375	69.23	121.28
105	516	93.97	154.14	470	87.63	150.96
120	621	111.68	182.26	580	106.91	183.40
135	726	129.56	211.09	705	126.74	216.50
150	831	147.61	239.83	840	147.14	250.26
165	936	165.79	269.12	975	168.02	282.32
180	1041	184.13	298.04	1110	189.21	315.35

Table 3.9: System-wide statistics on delay propagation: total propagated delay minutes

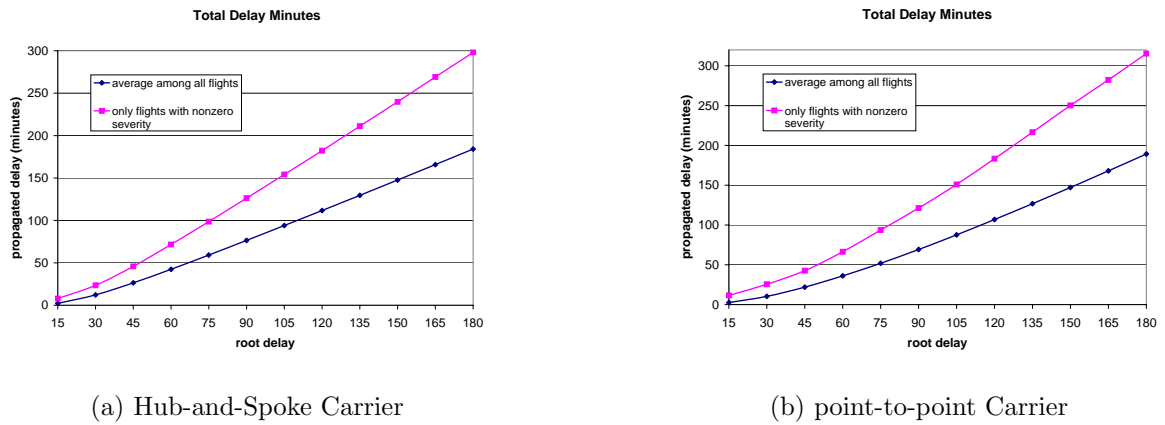
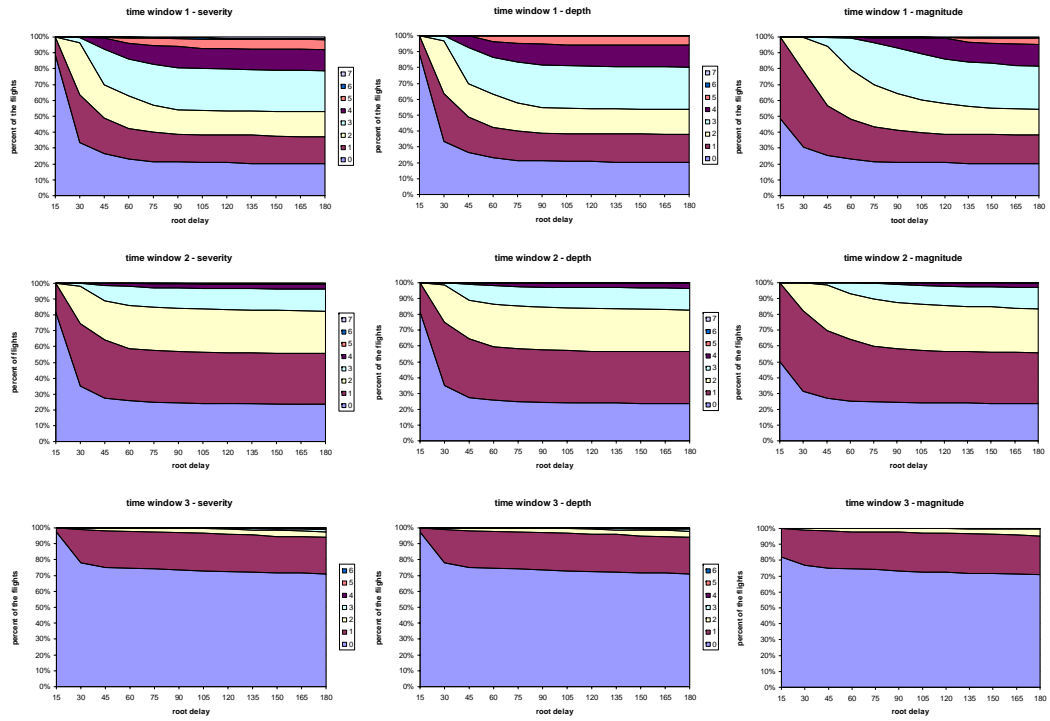


Figure 3.9: Average total propagated minutes of the propagation trees.

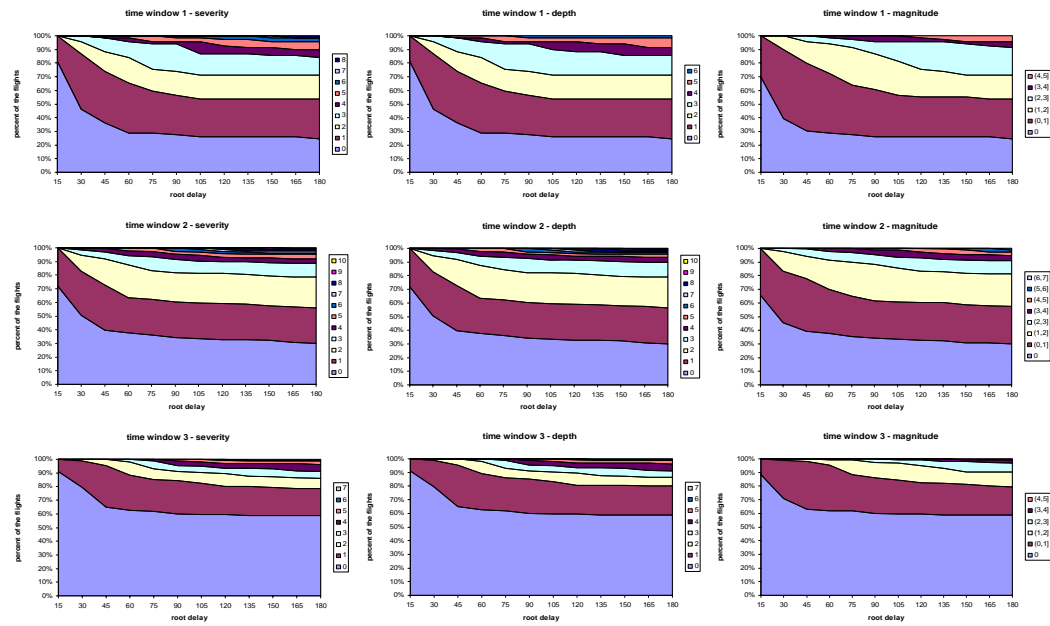
Time of Day In this section we examine whether time of day impacts the characteristics of the propagation trees. We start by dividing the flights into three categories based on their departure time:

1. departures between midnight and 8 AM.
2. departures between 8 AM and 4 PM.
3. departures between 4 PM and midnight.

We continue, as in our earlier analysis, to focus on individual root delays (i.e., we are not looking at interactions between flights, such as those occurring during peak



(a) Hub-and-Spoke Carrier



(b) point-to-point Carrier

Figure 3.10: Trends in the propagation measures categorized based on the departure time of the flights.

periods of congestion or times of severe weather disruption at a station). In this analysis, we are instead interested in whether the departure time of day, in and of itself, influences the structure of the propagation tree.

Although many carriers operate a limited number of “red-eye” flights overnight, the majority of flights typically occur between early morning (e.g. 6 AM) and late evening (e.g. 10 PM). In addition, crew members are frequently assigned to duties that start in the morning and end in the evening, with their rest periods occurring overnight, and scheduled aircraft maintenance is typically planned during the overnight period as well. Therefore, this introduces a natural “down time” in the system, in which any remaining delay propagations would typically be absorbed. As a result, it is reasonable to suspect that flights originating early in the day would have greater opportunity to propagate than flights originating later in the day. We consider this hypothesis in Figures 3.10(a) and 3.10(b).

In this figure, the top graphs show the trends in severity, depth, and magnitude of propagation trees for flights in the first time window (midnight to 8 AM); there are 266 (69) flights in this category. Similarly, the graphs in the middle row show the propagation tree characteristics for the flights in the second time window (8 AM to 4 PM); there are 908 (186) of these. Finally, the bottom row shows the characteristics of flights which depart during the third time window (8 PM to midnight); there are 545 (155) of these. Clearly, these figures demonstrate a strong relationship between the time of the original (root) delay and the degree to which that delay propagates.

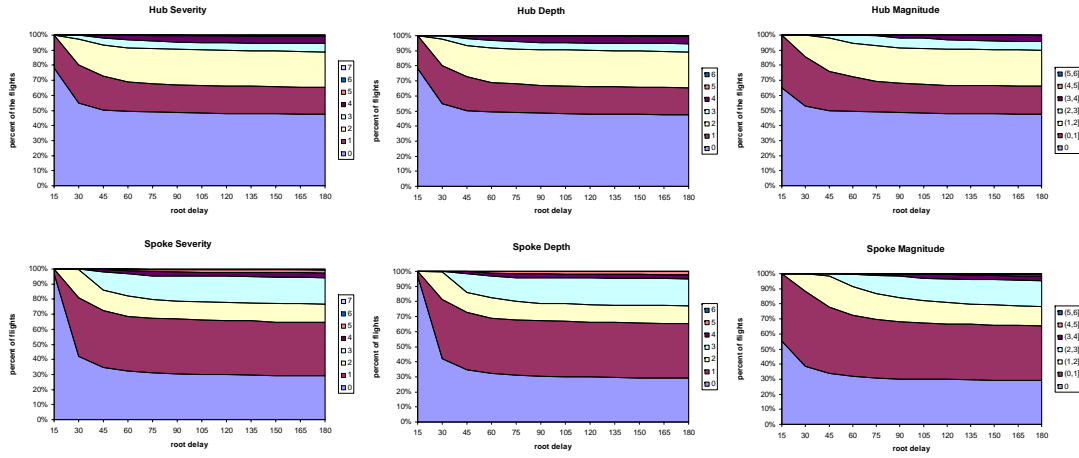
Of course, there are exceptions. The most prevalent of these are propagation trees that include red-eye flights spanning the overnight lull which typically absorbs residual delay. Nonetheless, the opportunities for delay propagation decrease significantly as departure time moves later in the day. For example, in the hub-and-spoke

network, if you look at the impact of a 180 minute root delay on the flights with the 100 latest departures, the average severity is 0.48 and the average number of propagated delay minutes is 35.08, whereas for the flights with the 100 earliest departures, the average severity is 2.48 and the average number of propagated delay minutes is 368.35.

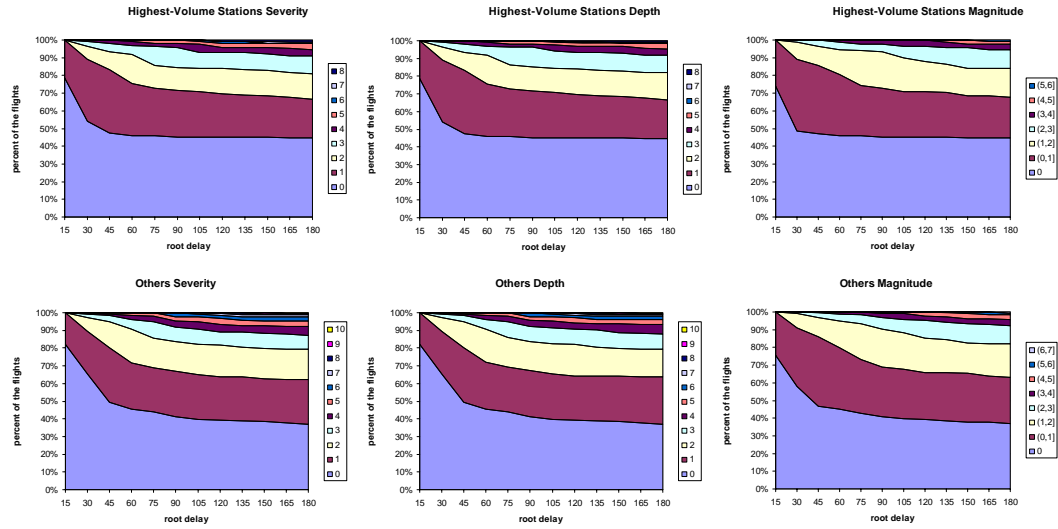
Similarly, in the point-to-point network, if you look at the impact of a 180 minute root delay on the flights with the 25 latest departures, the average severity is one and the average number of propagated delay minutes is 114, whereas for the flights with the 25 earliest departures, the average severity is 2.36 and the average number of propagated delay minutes is 304.88.

Hub vs. Spoke Another key characteristic of a flight, in addition to its departure time, is its origin airport. In particular, we question whether flights originating from hub vs spoke stations (or, in the case of the point-to-point carrier, the two highest-volume stations, which serve the majority of flights, vs the remaining stations) demonstrate different behaviors in their propagation trees. As demonstrated in Figures 3.11(a) and 3.11(b), this does in fact seem to be the case. Specifically, the *layers* of severity, depth, and magnitude are on average lower for the hub airports than for the other stations.

We theorize that this occurs as a combination of four factors. First, the majority of flights originate at a hub station, terminate at a hub station, or both. Second, as observed earlier, most propagation “trees” are in fact actually chains, with very limited splitting. As a result, the propagation tree associated with one flight might be virtually identical to the tree associated with its connecting flight, except for the addition of the original flight itself. Third, crews often start and end their pairings



(a) Hub-and-Spoke Carrier



(b) point-to-point Carrier

Figure 3.11: Trends in the propagation measures categorized based on the departure station size.

at hubs; similarly, aircraft often go out of rotation at hubs as well. Finally, there is often more slack between flights at the hubs (both to allow for connecting passengers and also because of the larger number of connecting opportunities), whereas sparse spokes often turn aircraft (and crews) as soon as possible.

As a result, if a flight from a hub to a spoke experiences a root delay, it will often experience a propagated delay at the spoke as well; in the more general sense, we commonly see even numbers of arcs in the tree. Conversely, delays to a flight originating at a spoke typically are either absorbed at the hub or else propagate for at least two more flights (hub to spoke and then spoke back to hub).

3.3.2 Observations

In the previous sections, we presented empirical results concerning the potential for root flight delays to propagate downstream in an airline network. In this section, we re-visit five commonly held viewpoints about delay propagation, and discuss these viewpoints in the context of our observations.

1. **Propagated delays create significantly more impact than the original root delays themselves.** Because of the interconnected use of multiple constrained resources, it is commonly assumed that the propagation of a delay in a flight network has greater impact than the root delay itself. On the one hand, our observations suggest that many flights do not, in fact, propagate root delays. Even with root delays of up to 180 minutes (and taking into account the caveats of Section 3.3.1), nearly 40% of the flights have no propagating effect. Furthermore, many flights that do propagate do so to a limited degree, impacting only one or two additional flights. On the other hand, this does not mean that flight delays do not propagate, nor that the impact is not of significance. In

particular, we note that for flights which *do* show delay propagation, for all but the briefest of root delay lengths (e.g. of less than 30 minutes), the magnitude for roughly half the flights is more than one, i.e., the propagated delay more than doubles the initial root delay, substantiating the importance of addressing delay propagation in network planning efforts.

2. **A single delay can “snowball” through the entire network.** The notion of a single flight delay propagating rampantly across the network, progressively expanding in its impact, is not in fact seen in our empirical observations. This may be partially attributed to the fact that we are not considering cabin crews and connecting passengers. [On the other hand, we are also not considering recovery decisions, instead assuming all delays propagate until absorbed.] More likely, the key “buffers” that have the greatest impact on limiting the propagation of delays are:

- crews going off-duty,
- crews and aircraft remaining together (and thus preventing one delay from causing two subsequent downstream delays),

and

- periods of decreased activity in the network (particularly in late evening, but also during lulls throughout the day).

Again, this does not suggest that delay propagation is not an important concern, but that these propagations are fairly localized and should be addressed correspondingly.

3. **Keeping aircraft and crews together can help to mitigate the impact**

of disruption. As suggested in the previous section, keeping aircraft and crews together appears to have tremendous benefit in reducing delay propagation. In particular, it implies that any one flight delay can lead to at most one subsequent delay, whereas splitting the crew and aircraft can potentially result in two downstream delays. It is interesting to note, however, that keeping crews and aircraft together is not sufficient to avoid delay propagation, and in fact, some of the flight delays with the maximum severity actually correspond to a propagation tree in which most flight connections do keep the crew and aircraft together. What is not clear from the data, and merits further study, is whether or not efforts to keep crews and aircraft together have any negative impact on delay propagation (i.e., lead to reduced amounts of slack between connecting flight pairs).

4. **Delays that occur early in the day can cause greater propagation than delays later in the day.** It seems logical that root flight delays early in the day will lead to greater propagation than root delays later in the day, because there are more opportunities for delay. This logic is premised on the notion, however, that there is a natural “break” at the end of the day which serves as a guaranteed buffer for any delays still propagating, and furthermore that such breaks do not occur at other times of the day. In the data that we considered this is predominantly, but not completely, true. For example, some flight delays do in fact propagate overnight, because the crews have short overnight rest periods - beginning their rest period late may force them to delay their first flight the subsequent day. In addition, there are times during the day where flight volumes decrease and thus early morning flight disruptions are absorbed into the system, rather than propagating through the day and into the evening

slack period. Nonetheless, we see a significant difference in delay propagation from one flight to another when factoring by departure time of day, with all three metrics (severity, depth, and magnitude) decreasing as the origin time of the root flight increases later into the day. Similar results have been reported in the literature by Beatty et al [8], Hsiao and Hansen [22] and Tu et al [55].

5. It is most important to prevent delay propagation early in the day.

The previous paragraph seems to support the conventional wisdom that it is most important to prevent delay propagations early in the day, i.e., given a fixed amount of slack in the system, that slack yields greater benefit when added to early morning flight connections. This logic, however, is premised on the notion that the disruption will actually occur. All other things being equal, a disrupted flight early in the day will in fact benefit more substantially from increased slack to absorb disruption than will a flight disrupted later in the day. On the other hand, slack early in the day also has less probability of being used. Consider, for example, the simple case where a crew and aircraft stay together for n consecutive flights, with zero slack between each flight pair. Suppose that we delay the first flight in the chain by five minutes. Then this delay will propagate until the end of the chain, resulting in an additional $5 * (n - 1)$ extra minutes of delay. This propagated delay could be eliminated by adding five minutes of slack after the first flight.

At the other extreme, consider a five minute delay to the second-to-last flight in the chain. This root delay would only lead to five minutes of additional propagated delay (associated with the final flight in the chain). Adding five minutes of slack to this connection would only save five minutes of propagated

delay, rather than $5 * (n - 1)$, again suggesting that adding slack earlier in the day is more beneficial than later in the day.

Now consider, however, the case where we add five minutes of slack to the first connection, but the first flight is *not* delayed. Then this slack would provide no benefit, and any subsequent root delay in the chain would propagate fully. On the other hand, putting the slack between the last pair of flights would save five minutes of propagated delay for *any* root delay in the chain. In fact, all other characteristics (and, in particular, the probability of a root delay) being equal, it can be shown that the optimal location for the slack is actually in the middle of the chain. This is the trade-off point where the expected delay is minimized, trading off the length of the propagation and the probability of the root delay. Of course, actual networks are much more complex. Thus, the question of where slack can most greatly benefit the network becomes a more challenging issue. This topic is further explored in Chapter IV.

3.4 Conclusions and Future Research

In this chapter, we have investigated the relationship between planned aircraft and crew schedules and the potential for delay propagation as these schedules are implemented. This research makes several important contributions towards understanding this relationship, a requisite precursor to developing more robust plans. First, we introduce the use of propagation trees as a visual and quantitative tool for assessing the ramifications of individual flight delays throughout the network. Second, we propose several metrics for quantifying the impact of these initial delays as they propagate through the network. Third, we use these metrics to conduct a quantitative analysis of planned airline schedules, gaining insights into the relationship between scheduled

slack and delay propagation. Fourth, we extend this analysis to substantiate (in some cases) and disprove (in other cases) commonly-held assumptions about delay propagation. Finally, we lay the groundwork for future research on incorporating measures of potential delay propagation in the airline schedule planning process. This future research includes the following four topics:

First, we see value in extending the complexity of this analysis – taking into account the probabilities of the occurrence of different root delays, recognizing correlations (for example, weather-based) between groups of root delays, adding in the propagation due to cabin crews and connecting passengers, and incorporating recovery decisions, such as crew swaps and flight cancelations.

Second, our research can be useful in helping to identify mechanisms for strategically using slack in the system to mitigate the impact of disruption, by recognizing where this slack can provide the greatest benefit. This idea is explored in Chapter IV.

Third, there is important work left to be done in quantifying the *value* of increased robustness – what is the trade-off between improved robustness (i.e., decreasing the likelihood of delay propagation) and increased scheduled costs (i.e., the cost of a plan if it operates without disruption)?

Finally, once the value of robustness can be quantified, it is possible to begin incorporating metrics of robustness within the planning process itself.

CHAPTER IV

Decreasing Airline Delay Propagation By Re-Allocating Scheduled Slack

4.1 Introduction

Airline plans are made up of several costly and constrained resources such as aircraft and crews. These resources link flights across the network, with each resource flowing from one flight to another. Adding to this complexity is the fact that although each flight needs each type of resource, individual resources do not necessarily stay linked throughout the network. For example, an aircraft and crew might be assigned to a common flight at a particular point in the schedule, but assigned to separate flights at a later point.

One ramification of this linkage is the potential for delays to propagate. If one flight is delayed (for example, because of a mechanical problem with the aircraft assigned to that flight), then a subsequent flight might also be delayed because it is awaiting that inbound aircraft. The fact that resources can “split” compounds this. In Chapter III, we discussed how a single flight delay can spread to delay several other flights as well.

There are many sources for flight delays, such as mechanical problems, weather delays, ground-hold programs, and air traffic congestion. But the secondary delays that propagate from such *root delays* are also quite substantial. For example, in

December 2007, more than one-third of the delays at major U.S. airports were the result of a late-arriving aircraft (Figure 1.3). Furthermore, there is a natural conflict stemming from the fact that slack is typically viewed as negative from the planning perspective (i.e., a waste of resources), but as positive from the operational perspective (i.e., an opportunity to absorb disruption rather than allowing it to propagate). The focus of our research is therefore on determining how to incorporate the operational issues associated with delay propagation into the airline planning process.

A key challenge in this research is the difficulty in trading off between *planned costs* (the cost of an airline plan under the assumption that all flights occur as scheduled and without disruption) and *operational costs* (the realized cost associated with the modified plan that is implemented in response to disruptions). Given two different plans with varying planned costs, it is difficult to determine which of the two plans will perform better operationally. Furthermore, it is also difficult to determine whether improvements in operational performance outweigh increases in planned costs, given that the plan will be operated several times (often, daily) over the planning horizon, and that potential disruptions may or may not occur during any given day. Thus, even determining metrics for “robustness”, and then quantifying the value of these metrics (i.e., how much planned cost a carrier should be willing to incur to improve these metrics), are challenging research topics themselves that have yet to be adequately solved. The fact that the planning and operations processes are functionally separate within most carriers, with each group’s incentives aligned with different objectives, only serves to exacerbate the problem.

We therefore propose, as an interim step, to develop an approach that does not increase planned costs, but can nonetheless improve operational performance. Specifically, we propose to modify flight departure times so as to re-allocate the existing

slack in the network. By re-timing flights, slack can be re-distributed to those flight connections that are most sensitive to disruption and thus delay propagation. We limit the time windows in which flights can be re-timed, so as to maintain existing revenue projections. Furthermore, we restrict flight re-timings such that crew pairings remain feasible and do not change in cost. Finally, we require that the same aircraft rotations be maintained. Our computational results, based on data from a major U.S. carrier, demonstrate that this approach leads to significant improvements in expected delay propagation without any associated increase in planned cost.

The primary contribution of this chapter is in developing models that can diminish delay propagation in operations, without any increase in planned costs. Our proposed models can take into account delay propagations caused by aircraft, crew members, connecting passengers, as well as other shared resources. By demonstrating that the integrality of these models can be relaxed (i.e., that the models can be solved as linear, rather than integer, programs), we are able to consider all down-stream impacts without sacrificing tractability. The chapter is outlined as follows: In Section 4.2 we review the related literature. We present models for re-allocating slack in Section 4.3, as well as a simulation model to help validate the results. Computational experiments are presented and analyzed in Section 4.4. Finally, Section 4.5 offers conclusions and suggested areas for future research.

4.2 Literature Survey

In this section, we first briefly review the different steps in the airline planning process. Then we focus on different approaches to addressing airline delays and present related literature.

4.2.1 Airline Planning Problems

The airline planning process is classically decomposed into four sub-problems: schedule generation, fleet assignment, maintenance routing, and crew scheduling.

The objective of the *schedule generation* problem is to determine what markets, frequencies, and times to fly in a given period of time, taking into account both forecasted demand data and available resources (Berge [9]). The *fleet assignment* problem determines which type of aircraft should be assigned to each flight, considering the demand and capacity constraints (Abara [1]). The *maintenance routing* problem is primarily a feasibility problem that assigns specific aircraft to flights to ensure adequate opportunities for required maintenance checks (Barnhart and Talluri [7]). The objective of the *crew scheduling* problem is to find the most cost-effective assignment of cockpit or cabin crews to flights (Barnhart et al [5]).

These airline planning problems are complex and large-scale by nature. Therefore, they are often treated as deterministic in order to achieve tractability, not taking into account the impact of delays and disruptions. In the following subsection, we highlight some of the approaches taken to incorporate disruptions in the airline planning process.

4.2.2 Robust Planning

Airline operations are subject to significant uncertainties. Disruptions often occur as a result of weather conditions, unplanned maintenance issues, safety checks, security concerns, and more. The goal of *robust planning* is to generate schedules that are less sensitive to these disruptions. This is a demanding task, given both the size and the uncertainty of the networks. There have been two major types of approaches to robust airline planning in the literature.

In the first approach, a stochastic programming model is used to capture uncertainties. Yen and Birge [59] developed a stochastic model for the airline crew scheduling problem and adopted a delay branching algorithm to solve the resulting problem. Alternatively, Rosenberger et al [45] used a simulation model to implement a stochastic approach for the crew scheduling problem and to evaluate recovery policies. Fuhr [18] developed a stochastic model to evaluate the on-time performance of a given schedule.

The second approach to modeling disruptions is to define surrogate problems that inherit the stochastic nature of the original problem. For example, Rosenberger et al [44] showed that fleet assignments with less hub connectivity and more short cycles perform better in operational circumstances. Schaefer et al [49] modeled the crew scheduling problem under uncertainty using approximated expected pairing costs. They also defined a lower bound on the expected cost of the pairings. Klabjan et al [28] defined a *regularity* measure as a way to capture robustness and considered maximizing this measure in addition to minimizing total cost.

4.2.3 Flight Re-timing Models

Perhaps the most closely related work to our research is that of Stojković et al [52] and Lan et al [33], both of whom also consider the use of flight re-timings to improve schedule performance.

In [52], the primary focus is on day-of-operations recovery activities. In particular, they focus on how to modify an existing plan in order to recover from a set of minor disruptions. They require that crew connections, rest requirements, aircraft connections, maintenance requirements, passenger connections, etc. all be maintained. They permit not only changes to flight departure times (specifically, increases), but also allow activities to be expedited. For example, the amount of time required to

off- and on-load passengers might be relaxed. The objective function then seeks to minimize the costs associated with extra resource utilization (for example, as needed when expediting activities) and passenger inconveniences.

Lan et al [33] consider two problems, both related to our research. First, they consider how changes in aircraft routings can be used to reduce the potential for delays to propagate via connecting aircraft. In this case, they hold flight departure times constant but allow the assignment of aircraft (i.e., tail numbers) to flights to change so as to better utilize the slack in the system to absorb disruption. In a separate problem, they keep the aircraft routings fixed but allow flight times to vary within a limited time window. The objective in this problem is to decrease the impact of delay on passengers' ability to make flight connections.

Both of these papers serve to demonstrate how even minor schedule modifications can have significant impact on system performance under disruptions, and help motivate our research.

The idea of using time windows in the airline planning context was first introduced by Levin [37]. Rexing et al [43] allowed scheduled flight departure times to vary within a given time window to improve flight connection opportunities and the cost-effectiveness of the fleet assignment. Stojković and Soumis [53] considered the problem of simultaneously modifying the existing flight departure schedule and planning individual work days (duties) while keeping planned aircraft itineraries unchanged. Mercier and Soumis [41] considered an integrated aircraft routing and crew scheduling model which allows the departure time of the flights to vary within a given time window. Burke et al [11] constructed a multi-objective optimization model to improve schedule reliability as well as schedule feasibility by re-timing flights and permitting minor changes in aircraft rotations while keeping the fleet assignments

fixed.

We seek to extend this research by developing a flight re-timing model that focuses on minimizing the propagation of root flight delays. Our approach permits the simultaneous consideration of interactions between multiple resources (aircraft, crews, passengers, etc.) and follows the downstream propagation of delays until absorbed. In particular, we are able to do so in a linear (rather than integer) program, which has significant benefits in terms of computational performance. Because our model relies on a surrogate objective function to approximate delay propagation, we also develop a simulation-based approach to mimic the propagation of delays operationally. This simulation assists us in assessing the quality of our surrogate objective function.

4.3 Models for Re-Allocating Planned Slack

4.3.1 Main Idea

Our goal is to improve the expected *operational performance* of a planned airline schedule without increasing its *planned cost*. In particular, we want to re-time flight departures so as to re-distribute existing *slack* (the scheduled connection time between two flights sharing a common resource minus the minimum turn time between these flights) in the network, to make this slack available where it is most needed operationally. Note that by moving a flight’s departure earlier, we increase the slack in its outbound connections, but decrease the slack in its inbound connections (see Figure 4.1). Therefore, given a fixed amount of slack in the schedule, we want to re-distribute this slack to where it can best be utilized, taking into account both the current connection times (e.g. adding slack to a long connection is unlikely to provide significant benefit) and also the likelihood of root delays, which determine how frequently the slack will be needed.

We limit the extent to which each flight can be re-timed in three ways. First, we maintain the feasibility of the existing crew and aircraft assignments. Second, we protect connecting passenger itineraries. Third, we do not permit flight times to be modified so substantially that the projected demand levels would change. Although these restrictions limit the extent to which the schedule can be improved, we are nonetheless able to provide an interim level of improvement (which we show to be non-trivial) that can be achieved immediately, while researchers continue to study the broader challenges of how to quantify, value, and increase schedule robustness through more drastic changes.

The decisions in our model are how much earlier or how much later to re-time each flight’s departure. The constraints needed to enforce the limitations on how flights can be re-timed (so as to meet the three restrictions outlined previously) are fairly straightforward, as is seen in the following sections. The challenge lies in determining how to represent the objective function, so as to maintain tractability while providing a solution that does in fact reduce delay propagation.

To explain our objective function (which is a surrogate for “robustness”), first consider a single flight f . For this flight, we also have a set of relevant connections. For example, the aircraft assigned to flight f might next connect to flight g , while the cockpit crew connects to flight h . In addition, the cabin crew might connect to flight i , and flights j and k may represent key passenger connections. Each of these connections has some (non-negative but possibly zero) scheduled slack. In our constraints, we enforce that the change in departure time for flight f does not violate the minimum turn times for these connections – in other words, this scheduled slack must remain non-negative.

But what happens if flight f is delayed in operations? For example, suppose its

assigned aircraft has a mechanical problem that takes 35 minutes to repair. In the absence of a recovery intervention, any available slack will be used to absorb this delay, with the *residual delay* (the delay beyond the available slack) propagating to the connecting flights. In our example, if the slack between flights f and g is 40 minutes, then flight g will not be delayed, but if the slack between flights f and h is only 20 minutes, then h will experience a 15 minute departure delay. If we move the departure time of flight f to be 15 minutes earlier, then a 35 minute root delay to flight f will no longer propagate to impact flight h .

As a surrogate objective function, we propose to consider the sum of the impacts of each individual root delay as it propagates downstream, weighting these delays by their relative probabilities. There are of course limitations to this approach. First, we are not considering recovery decisions, which can impact how delays propagate. Incorporating recovery is a sizeable challenge, largely due to the fact that recovery decisions are not pre-defined but rather based on individual personnel's prior experience and intuition. We nonetheless suggest that incorporating some baseline rules for recovery interventions would be an important extension of our model to consider. Second, we do not consider the interactions of root delays and, as a result, we over-count propagation. For example suppose that the aircraft of flight f connects to flight g , while the crew of flight h also connects to flight g . When we consider the impact of a root delay to flight f , we may capture a downstream impact on flight g . The same will occur when we consider the impact of a root delay to flight h . In reality, should these two root delays occur concurrently, then their effect on flight g will typically *not* be additive. Of course, this over-counting will occur in both the original schedule and our proposed alternative.

While these limitations will impact the quality of our results, we nonetheless

suggest that our surrogate function can improve over the existing schedule without increasing planned costs, enabling carriers to see immediate improvements in their operational performance while the research community continues to seek ways to provide further benefits. To support this hypothesis, we have developed a discrete-event simulation model (presented in Section 4.3.4) to help assess the quality of our proposed solutions.

4.3.2 Single-Layer Model

We begin by presenting a *single-layer model (SLM)* for redistributing slack. This model only considers the impact of disruptions one layer “downstream” - that is, on flights directly connected to the flight experiencing the root disruption.

We present this model for two reasons. First, it provides a simple framework that will facilitate understanding of the more complex multi-layer model. Second, as demonstrated in Section 4.4, even this simple model can yield non-trivial benefits.

Notation

Sets

\mathcal{F}	set of flights
\mathcal{A}	set of all considered connections
\mathcal{M}	set of possible delay values (minutes)
	- here we assume that \mathcal{M} is a discrete set

Parameters

$k_f^+ \geq 0$	$\forall f \in \mathcal{F}$	the amount by which the departure time of flight f can be moved later
$k_f^- \geq 0$	$\forall f \in \mathcal{F}$	the amount by which the departure time of flight

	f can be moved earlier
$0 \leq p_f^m \leq 1 \quad \forall m \in \mathcal{M}, \forall f \in \mathcal{F}$	the probability that flight f experiences a root delay of m minutes
$s_{f_1, f_2} \geq 0 \quad \forall (f_1, f_2) \in \mathcal{A}$	the slack between flights f_1 and f_2 in the original schedule.

Decision Variables

$$\begin{aligned}
 -k_f^- &\leq x_f \leq k_f^+ && \forall f \in \mathcal{F} \\
 y_{f_1, f_2} &\geq 0 && \forall (f_1, f_2) \in \mathcal{A} \\
 d_{f_1, f_2}^m &\geq 0 && \forall (f_1, f_2) \in \mathcal{A}, \forall m \in \mathcal{M}
 \end{aligned}$$

x_f is the change in the departure time of flight f . This change is restricted to the specified range $[-k_f^-, k_f^+]$. Note that a negative value for x means the flight is shifted earlier and a positive value for x means that the flight is shifted later.

y_{f_1, f_2} corresponds to the new slack between flights f_1 and f_2 , according to the modified schedule.

d_{f_1, f_2}^m corresponds to the delay that will propagate from flight f_1 to flight f_2 in the new schedule if there is a root delay of m minutes imposed on flight f_1 .

Formulation

$$(4.1) \quad (\text{SLM}) \quad \min \sum_{m \in \mathcal{M}} \sum_{(f_1, f_2) \in \mathcal{A}} p_{f_1}^m d_{f_1, f_2}^m$$

s.t.

$$(4.2) \quad y_{f_1, f_2} = s_{f_1, f_2} - x_{f_1} + x_{f_2} \quad \forall (f_1, f_2) \in \mathcal{A}$$

$$(4.3) \quad d_{f_1, f_2}^m \geq m - y_{f_1, f_2} \quad \forall (f_1, f_2) \in \mathcal{A}, \forall m \in \mathcal{M}$$

$$(4.4) \quad d_{f_1, f_2}^m \geq 0 \quad \forall (f_1, f_2) \in \mathcal{A}, \forall m \in \mathcal{M}$$

$$(4.5) \quad -k_f^- \leq x_f \leq k_f^+ \quad \forall f \in \mathcal{F}$$

$$(4.6) \quad y_{f_1, f_2} \geq 0 \quad \forall (f_1, f_2) \in \mathcal{A}$$

The objective function (4.1) of SLM minimizes the expected value of the delay propagation by weighting the probability of an m -minute delay on flight f_1 times its propagation to f_2 , summed over all flight connections (f_1, f_2) and delay lengths m . [Recall that, in this objective function, we consider only one layer of propagation.]

Constraints (4.2) calculate the new slack (y_{f_1, f_2}) between two flights. This is the old slack (s_{f_1, f_2}) minus the change in the departure of the first flight (x_{f_1}) , plus the change in the departure of the second flight (x_{f_2}) . Note, as illustrated in Figure 4.1, that if flight f_1 is moved earlier, then x_{f_1} will have a negative value and the slack y_{f_1, f_2} will therefore *increase* because we are subtracting this value. As a result, the new slack (y_{f_1, f_2}) equals the old slack (s_{f_1, f_2}) minus x_{f_1} plus x_{f_2} . In the example given in Figure 4.1, the departure of the first flight is shifted earlier by x_{f_1} (which is negative) and the departure time of the second flight is pushed forward by x_{f_2} (which is positive). Constraint set (4.6) ensure that the connection stays feasible, i.e., that the slack remains non-negative.

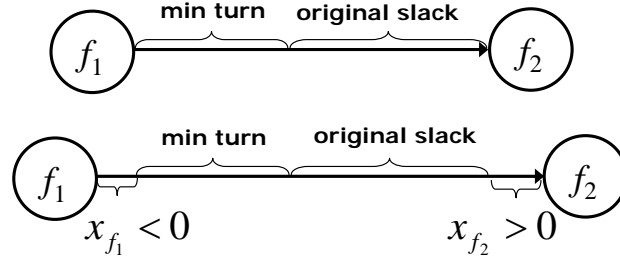


Figure 4.1: Visualizing constraint (2).

Constraint sets (4.3) and (4.4) calculate how much delay would propagate from flight f_1 to its outbound connection f_2 if f_1 were to experience a root delay of m minutes. Specifically, the propagated delay is m minus y_{f_1, f_2} (the root delay minus the new slack between the flights) unless this is negative, in which case the delay is zero.

Finally, constraints (4.5) limit the amount by which flight times can be changed. Note that this is flight-specific and can be used not only to restrict changes so that market share is not impacted, but also to recognize gate limitations, slot restrictions, hourly departures (in which case the time window would be zero), etc. Key passenger itineraries could be protected as well, by placing a lower value on the slack time (y) between two connecting flights.

Claim: The solution to (4.2)-(4.6) will yield integer values of x .

Proof. In order to prove the claim, we need to show that the coefficient matrix presented by (4.2)-(4.6) is totally unimodular. In that case, given that all elements in the right-hand-side vector of (4.2)-(4.6) are integer, all the extreme points of (4.2)-(4.6) will be integer [42].

In order to show that the coefficient matrix defined by (4.2)-(4.6) is totally unimodular, it is sufficient to prove that the coefficient matrix corresponding to constraints (4.2) and (4.3) – or equivalently, its transpose – is totally unimodular. The

constraints (4.4)-(4.6) are upper and lower bounds, which do not impact the claim.

We can re-write constraint (4.3) substituting (4.2), which yields:

$$(4.7) \quad d_{f_1, f_2}^m \geq m - s_{f_1, f_2} + x_{f_1} - x_{f_2} \quad \forall (f_1, f_2) \in \mathcal{A}, \forall m \in \mathcal{M}$$

By transferring the variables to the left-hand-side we get:

$$(4.8) \quad d_{f_1, f_2}^m - x_{f_1} + x_{f_2} \geq m - s_{f_1, f_2} \quad \forall (f_1, f_2) \in \mathcal{A}, \forall m \in \mathcal{M}$$

Here we can see that the coefficient matrix (A) corresponding to constraint (4.8) has entries (a_{ij}) of only -1, 0 and 1. Furthermore, we can partition the columns of this matrix into two sets: \mathcal{C}_1 includes the columns corresponding to the d variables and \mathcal{C}_2 the columns corresponding to the x variables. We see that for each row of this matrix, the summation of all the coefficients in \mathcal{C}_1 equals 1 and the summation of all the coefficients in \mathcal{C}_2 equals 0. Therefore:

$$(4.9) \quad \left| \sum_{j \in \mathcal{C}_1} a_{ij} - \sum_{j \in \mathcal{C}_2} a_{ij} \right| \leq 1 \quad \forall i$$

Therefore, A' (the transpose of A) is totally unimodular, and thus A itself is also totally unimodular. \square

This claim tells us that it is sufficient to solve SLM as an LP, rather than an IP, while still yielding integer departure times. This has significant implications not only for the tractability of the model, but also for our ability to extend it to include indirect downstream effects, as we see in the next section.

4.3.3 Multi-Layer Model

The model presented in Section 4.3.2 only considers the impact of a root delay on the flight's immediate connections. Of course, these delayed connections can in turn

delay their outbound connections as well if there is not enough slack to fully absorb the disruption. In this section, we present a *multi-layer model (MLM)* in which the downstream propagation of delays continues until they are fully absorbed.

To formulate this model, we must formalize the notion of a *worst-case propagation tree*. Note that in Chapter III, we formally defined propagation trees as the set of flights which can be delayed as a result of an m -minute delay in root flight f_0 . Here, because we construct these trees *before* re-timing the flights, we do not know for sure which flights will receive propagated downstream delays. Therefore the propagation tree considers the “worst-case” scenario. For each root flight f_0 and each delay value m , the worst-case propagation tree represents the set of all downstream flights that could potentially be delayed as a result of the root delay propagating. Note that in the remainder of this section, the term “propagation tree” corresponds to the worst-case propagation trees.

To construct a worst-case propagation tree for root flight f_0 and delay value m , we look at each of its outbound connections. For example, let f_1 be the flight awaiting flight f_0 's aircraft. We assume that f_0 is moved to its latest permitted departure time and f_1 is moved to its earliest departure time – this creates the minimum possible slack between each flight pair. Clearly, if a delay of length m does not propagate across this slack, then it will not propagate across any re-timing of the flight pair. On the other hand, if it does, then the connecting flight is added to the propagation tree. For each outbound connection that is added to the propagation tree, we then look at all of its outbound connections, again assuming the minimum amount of slack in the connection. Observe that for a given root flight f_0 , the propagation tree for a delay of length m_1 will be a subset of the propagation tree for the delay of length $m_2 > m_1$. Figure 4.2 presents two such propagation trees, where the potential propagation from

f_i to f_j is calculated as f_i 's root delay minus the slack between f_i and f_j plus the maximum amount by which the departure time of f_i can be moved later ($k_{f_i}^+$) plus the maximum amount by which the departure time of f_j can be moved earlier ($k_{f_j}^-$). In this figure, the propagation tree with root flight f_1 and root delay 30 ($\mathcal{T}_{f_1}^{30}$, above) has fewer nodes compared to the propagation tree with root flight f_1 and root delay 60 ($\mathcal{T}_{f_1}^{60}$, below).

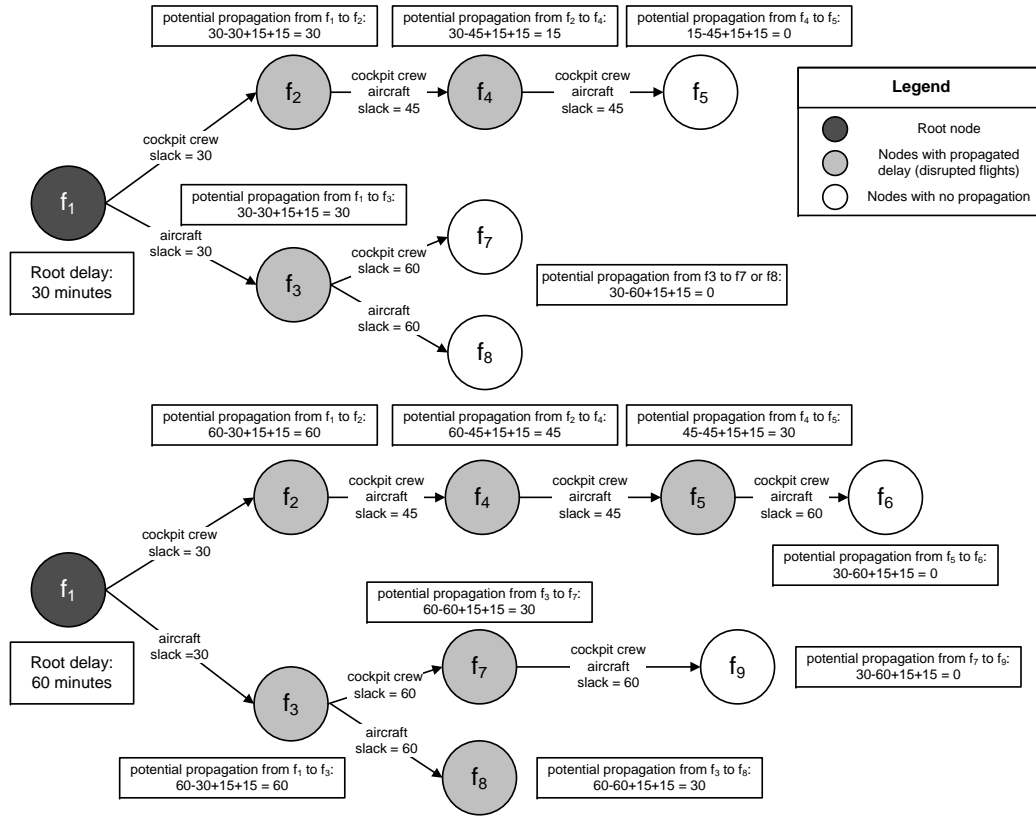


Figure 4.2: Visualizing *worst-case* propagation trees.

We define the following notation:

- $\mathcal{T}_{f_0}^m$ set of flights in the worst-case propagation tree associated with root flight f_0 and root delay m (excluding f_0)
- $r_{f_0}^m(f)$ the parent node of flight f in $\mathcal{T}_{f_0}^m$.

Based on this notation, we extend the single-layer model to the multi-layer model:

$$\begin{aligned}
(4.10) \quad (\text{MLM}) \quad & \min \sum_{m \in \mathcal{M}} \sum_{(f_0 \in \mathcal{F}, f \in \mathcal{T}_{f_0}^m)} p_{f_0}^m d_{f_0, f}^m \\
& \text{s.t.} \\
(4.11) \quad & y_{f_1, f_2} = s_{f_1, f_2} - x_{f_1} + x_{f_2} && \forall (f_1, f_2) \in \mathcal{A} \\
(4.12) \quad & d_{f_0, f}^m \geq m - y_{f_0, f} && \forall (f_0 \in \mathcal{F}, f \in \mathcal{T}_{f_0}^m : r_{f_0}^m(f) = f_0), \forall m \in \mathcal{M} \\
(4.13) \quad & d_{f_0, f}^m \geq d_{f_0, r_{f_0}^m(f)}^m - y_{r_{f_0}^m(f), f} && \forall (f_0 \in \mathcal{F}, f \in \mathcal{T}_{f_0}^m : r_{f_0}^m(f) \neq f_0), \forall m \in \mathcal{M} \\
(4.14) \quad & d_{f_0, f}^m \geq 0 && \forall (f_0 \in \mathcal{F}, f \in \mathcal{T}_{f_0}^m), \forall m \in \mathcal{M} \\
(4.15) \quad & -k_f^- \leq x_f \leq k_f^+ && \forall f \in \mathcal{F} \\
(4.16) \quad & y_{f_1, f_2} \geq 0 && \forall (f_1, f_2) \in \mathcal{A}
\end{aligned}$$

There are two key differences between the single-layer model presented in Section 4.3.2 and this model. First, in the objective (4.10), we include not only the delay minutes that a root delay of m minutes on flight f_0 imposes on f_0 's immediate outbound connections, but also the propagated impact on all subsequent flights in f_0 's propagation tree.

Second, constraints (4.13) enforce the fact that the delay propagated to a flight downstream from the root delay will be the amount of delay propagated to its parent minus the amount of slack between these flights.

This model is structurally quite similar to the single-layer model and exhibits the same integrality property. The main difference is in size. For each flight in any given propagation tree, we have to add both a new variable and a new constraint. As we observe in Section 4.4, however, the modified formulation remains highly tractable for networks of a realistic size.

Claim: The solution to (4.11)-(4.16) will yield integer values of x .

Proof. The proof is similar to the one presented in 4.3.2. Here we argue that the coefficient matrix presented by (4.11)-(4.13) is totally unimodular. (The constraints (4.14)-(4.16) again do not affect the integrality of the x variables.)

As in the earlier proof, we can re-write constraints (4.12) by substituting in constraints (4.11), yielding:

$$(4.17) \quad d_{f_0,f}^m - x_{f_0} + x_f \geq m - s_{f_0,f} \quad \forall (f_0 \in \mathcal{F}, f \in \mathcal{T}_{f_0}^m : r_{f_0}^m(f) = f_0), \forall m \in \mathcal{M}$$

Note that this constraint, which captures the relationship between the root flight and each of its “children”, includes one d variable, with coefficient of 1, and two x variables, one with coefficient 1 and one with coefficient -1.

Next we consider constraint set (4.13). First, suppose that f is a “grandchild” of f_0 – in other words, its inbound connection is f_0 ’s outbound connection. Then by substituting (4.11) we get:

$$(4.18) \quad d_{f_0,f}^m \geq d_{f_0,r_{f_0}^m(f)}^m - s_{r_{f_0}^m(f),f} + x_{r_{f_0}^m(f)} - x_f$$

and then by substituting (4.12) we get:

$$(4.19) \quad d_{f_0,f}^m \geq m - s_{f_0,r_{f_0}^m(f)} + x_{f_0} - x_{r_{f_0}^m(f)} - s_{r_{f_0}^m(f),f} + x_{r_{f_0}^m(f)} - x_f$$

Canceling and moving the variables to the left hand side of the constraint yields:

$$(4.20) \quad d_{f_0,f}^m - x_{f_0} + x_f \geq m - s_{f_0,r_{f_0}^m(f)} - s_{r_{f_0}^m(f),f}$$

which again has one d variable with coefficient one and two x variables, one with coefficient 1 and one with coefficient -1.

More generally, when f is a descendant of f_0 in the propagation tree, substitution as above will yield:

$$(4.21) \quad d_{f_0,f}^m - x_{f_0} + x_f \geq m - \sum_{(f_1,f_2) \in \mathcal{P}_{f_0}^m(f)} s_{f_1,f_2}$$

as all the intermediate x variables cancel, with each intermediate flight on the path from f_0 to f playing the role of both parent and child. [In equation 4.21, $\mathcal{P}_{f_0}^m(f)$ is the set of all arcs (f_1, f_2) that form the path from the root node f_0 to f in the propagation tree $\mathcal{T}_{f_0}^m$.] Again, we see one d variable with coefficient 1 and two x variables, one with coefficient 1 and one with coefficient -1.

Thus, using the same partitioning of the columns of the A matrix as the earlier proof, the sufficient condition for total unimodularity follows directly. \square

4.3.4 Simultaneous Delays Model

The models presented in Sections 4.3.2 and 4.3.3 rely on the use of a surrogate objective function to achieve robustness. In particular, objective functions (4.1) and (4.10) attempt to minimize the total amount of propagated delay in the flight network by looking at how each individual flight propagates delay. The primary limitation of this approach is that it does not take into account the fact that multiple flight delays may occur in the network simultaneously. As a result, propagated delay will in some cases be estimated inaccurately. Figure 4.3 presents one such situation.

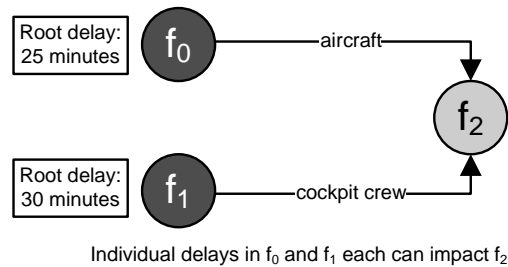


Figure 4.3: An example of simultaneous root delays where the surrogate objective function *overestimates* the delay propagations.

Given flights f_0 and f_1 , both of which are parent (preceding) flights of f_2 , the

objective function includes the sum of the delays propagated from both f_0 and f_1 . In scenarios where both f_0 and f_1 experience a root delay simultaneously, however, f_2 's delay should be the *maximum* of the two upstream delays, rather than their sum.

Conversely, there are occasions where the surrogate objective function underestimates delay. For example, as illustrated in Figure 4.4, when simultaneous root delays happen consecutively, their overall results can be more disruptive than what the objective functions in (4.1) or (4.10) can estimate. In this figure, individual root delays in f_0 or f_1 are not enough to delay f_2 . However, when f_0 and f_1 both experience a thirty-minute delay, their delays *can* propagate to cause a ten-minute delay in f_2 .

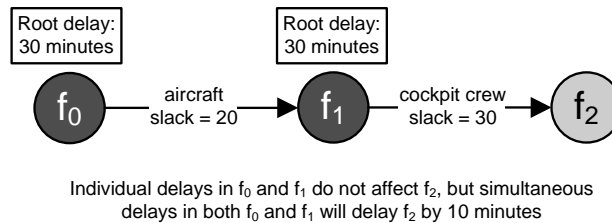


Figure 4.4: An example of simultaneous root delays where the surrogate objective function *underestimates* the delay propagations.

Given that both the original and the re-timed schedules suffer from these inaccuracies, it is not clear how the overall reduction in delay propagation is affected. Therefore, we have developed a discrete-event simulation model to provide further analysis into the overall effects of this approximation. This enables us to compare how both the original and the optimized schedules perform when multiple flights in the network incur root delays concurrently.

Note that we assume root delays to be additive, meaning that if a flight has previously incurred a propagated delay, but additionally incurs a root delay, the total departure time is pushed back by the sum of these values. For example, consider the

case when a flight is delayed because it is awaiting its (delayed) inbound aircraft. A weather delay or mechanical delay (i.e., a root delay) at the second flight would typically not arise until after the incoming aircraft arrived, making these delays additive. On the other hand, when two different upstream delays simultaneously affect a flight (e.g. its crew is delayed on one inbound flight and its aircraft delayed on another), then the propagation will be the maximum of the two propagated delays, rather than their sum.

To accurately simulate a given flight network using a discrete-event simulation, we require a source of randomness that will be used to determine the amount of root delays incurred by each flight. To model these delays, we have generated empirical distributions based on the origin airport of the departing flight, using actual delay data spanning a 12-month period. Specifically, we filtered out all delays that were propagated from an upstream root disruption. We then clustered the remaining delays by origin station and length of disruption. The corresponding probability mass functions were used in both the simulation, to randomly generate the initial root delays, and as coefficients in the objective functions of SLM and MLM.

Note that in a given flight schedule, there is no obvious “first flight” – because the schedule is repeating, any particular flight can appear as both the root of one propagation tree and a downstream flight in another. To address this complication, our simulation algorithm employs a recursive strategy that explores all flights in a given network without requiring a sequential ordering [34].

The delay that a flight experiences consists of two parts – propagated delay (resulting from the need to wait for delayed upstream resources such as cockpit crews, cabin crews and aircraft) and root delay (associated with the flight itself, such as a weather delay). The propagated delay is computed by taking the maximum delay

across all of the inbound resources, then adding this to the root delay.

To compute the total propagated delay in the network, we first initialize all flights in the network to have a propagated delay of zero and a root delay of zero. After this initialization stage, the propagation algorithm proceeds by executing the following steps for each flight in the network.

We pick an arbitrary flight in the network to process and call upon the random delay generator to provide a root delay for this particular flight. If this root delay is non-zero, we update the departure time of the current flight to be the sum of the (current value of the) propagated delay and this additional root delay. We then consider all outbound connections from this particular flight. For each connection, we determine how much (if any) of the current flight’s delay would propagate to this child. We then check whether this propagated delay is larger than the connecting flight’s current propagated delay. If it is, we update the connecting flight’s propagated delay. Next, we add this to the connecting flight’s root delay (which will be zero if that flight hasn’t been processed yet) to determine its new departure time. Finally, we recursively use this new departure time to update the propagated delay of its outbound connections, repeating until a flight delay does not propagate.

This process repeats until all flights in the network have been processed and therefore exposed to the possibility of incurring a root delay. Through the recursive nature of the algorithm, we are guaranteed to explore every flight and every propagation tree, updating them with the appropriate amount of propagated delay.

4.4 Computational Experiments

The objectives of our computational experiments were three-fold. First, we wanted to assess the run-time performance of our models and determine their tractability.

Second, we wanted to evaluate the extent to which minor changes in flight times could impact the potential for delays to propagate – would the optimized schedule have significantly less delay propagation than the original schedule? Third, we wanted to use simulation to assess the accuracy of our surrogate objective function, i.e., our metric for delay propagation. When multiple delays are allowed to occur simultaneously, as is the case in reality, does the our new schedule still show improved performance over the original schedule?

Our computational experiments were conducted using data provided by a major U.S. carrier offering over 500 flights per day. We considered two different dates in 2007. For each of these dates, we were given the flight schedule, cockpit crew schedules, and aircraft rotations. Thus, we were able to consider propagation of delays associated with incoming aircraft and cockpit crews. [We did not include delays associated with cabin crews or connecting passengers, due to lack of data.]

As a default, we assumed that each flight was allowed to be moved up to fifteen minutes earlier or fifteen minutes later than its original departure time. Note that, in theory, this could impact crew costs and feasibility. For example, if a duty were currently at its maximum allowed elapsed time and we moved the departure time of the first flight of the duty earlier and/or moved the departure time of the last flight of the duty later, then we would violate the elapsed time limit.

To account for this, we considered four different variations. In the first and most restrictive case, we assumed that the first flight of any duty could not be moved earlier and the last flight of any duty could not be moved later. This greatly limits the flexibility of the network. In both of the data sets considered, most of the flights are either the first or last flights of their duty, and thus only about twenty-five percent of the flights were allowed to move freely. Furthermore, this overly restricts

the system: in a duty whose cost is dominated by flying time, increasing the elapsed time by stretching the first and last flights slightly may have no impact on cost, and the feasibility of the duty may be unchanged as well.

Therefore, we considered three additional instances, each progressively less restrictive. In these, the first and last flights of the duty were only allowed to change by five minutes, by ten minutes, or by the same fifteen minutes as all other flights in the network.

All code was implemented and run on an Intel[®] Pentium[®] D 3.20 GHz CPU architecture using the C++ programming language. The optimization models were developed using CPLEX/Concert Technology and solved using CPLEX 11.0 solver.

4.4.1 Goal 1 – Tractability

Table 4.1 shows the size and run time of each problem instance solved. The first column of this table indicates which of the two data sets is being considered. The second column indicates the restriction on first and last flights in a duty (note that all other flights have a \pm fifteen minute time window in all instances). The third column indicates whether the single-layer or multi-layer model is being used. The fourth, fifth, and sixth columns give the size of the model (in number of constraints, number of variables and number of nonzero elements) and the final column gives the run time (in seconds).

Note that all run times are less than 10 seconds, suggesting that there is no computational limitation on either model. Although the model size increases substantially when moving from the single-layer model to the multi-layer model, the fact that the problem can be solved as an LP implies that tractability is not sacrificed when using the more accurate model.

Finally, these results suggest that even if a significant number of additional con-

Data Set	Duty Restriction	Model	No. of Constraints	No. of Variables	No. of Nonzeros	Run Time
1	0	SLM	1,319	1,592	2,869	4
1	5	SLM	2,278	2,724	4,938	3
1	10	SLM	2,498	2,975	5,420	4
1	15	SLM	2,701	3,217	5,867	4
1	0	MLM	4,745	5,048	12,328	7
1	5	MLM	5,874	6,356	15,626	7
1	10	MLM	5,995	6,494	15,853	7
1	15	MLM	6,108	6,636	16,080	8
2	0	SLM	1,041	1,251	2,245	3
2	5	SLM	2,081	2,505	4,509	3
2	10	SLM	2,269	2,714	4,906	3
2	15	SLM	2,444	2,923	5,289	2
2	0	MLM	3,932	4,178	10,035	6
2	5	MLM	5,131	5,573	13,526	7
2	10	MLM	5,273	5,733	13,835	6
2	15	MLM	5,376	5,865	14,081	6

Table 4.1: Size of the instances and their corresponding run times.

nections were considered (e.g. incorporating cabin crews, key passenger itineraries, etc), run times would remain tractable.

4.4.2 Goal 2 – Impact

Our second goal was to evaluate the potential improvements in delay propagation to be gained by re-timing flights slightly. First, we began by computing two estimates of the propagated delay of the original schedules – one using the objective function from the single-layer model and one using the more accurate objective function from the multi-layer model. We then optimized the original schedules twice, once using each of the two objective functions. Again, for each of these solutions, we computed both the SLM and MLM objective estimates of delay propagation.

Tables 4.2 and 4.3 presents these results. The first column specifies which of the two data sets is being analyzed. The second column indicates how many minutes the first/last flights of a duty are allowed to change. The next three columns present the objective function value of the original, single-layer optimal, and multi-layer optimal schedules relative to the single-layer objective function. For the optimized schedules,

($n\%$) indicates the percent improvement over the original schedule, relative to the single-layer objective function. Likewise, the last three columns in Table 4.3 represent the multi-layer objective function applied to the three schedules.

Data Set	Duty Restr.	Evaluated w/ SLM Obj. Function		
		Orig. Sch.	SLM Opt. Sol.	MLM Opt.Sol.
1	0	1,294.24	1212.89 (6.3%)	1231.07 (4.9%)
1	5		996.102 (23.0%)	1018.69 (21.3%)
1	10		859.65 (33.6%)	882.99 (31.8%)
1	15		756.49 (41.5%)	771.08 (40.4%)
2	0	1,192.81	1129.54 (5.3%)	1139.59 (4.5%)
2	5		912.88 (23.5%)	929.98 (22.0%)
2	10		782.05 (34.4%)	800.21 (32.9%)
2	15		678.61 (43.1%)	691.48 (42.0%)

Table 4.2: Reduction in the expected delay propagation – comparing the original schedule schedules with SLM.

Data Set	Duty Restr.	Evaluated w/ MLM Obj. Function		
		Orig. Sch.	SLM Opt.Sol.	MLM Opt. Sol.
1	0	2,268.55	2156.22 (5.0%)	2104.95 (7.2%)
1	5		1694.16 (25.3%)	1647.24 (27.4%)
1	10		1380.14 (39.2%)	1340.32 (40.9%)
1	15		1152.08 (49.2%)	1112.17 (51.0%)
2	0	2,037.05	1944.77 (4.5%)	1919.01 (5.8%)
2	5		1518.1 (25.5%)	1481.76 (27.3%)
2	10		1224.79 (39.9%)	1192.33 (41.5%)
2	15		1008.55 (50.5%)	973.816 (52.2%)

Table 4.3: Reduction in the expected delay propagation – comparing the original schedule schedules with MLM.

Observe that, as expected, the amount of propagation is larger for all scenarios when multiple layers of propagation are taken into account. In addition, the single-layer optimal solution of course performs better under the single-layer objective, while the multi-layer optimal solution performs better under the multi-layer optimization.

It is interesting to note that there is not a dramatic difference between the single- and the multi- layer schedules in their performance under the multi-layer objective function. This is presumably due to the fact that the delays dissipate fairly quickly as observed Chapter III and thus the immediate outbound connection plays a dominant role. Therefore, minimizing the first layer of delays will capture much of the possible

benefits.

Another interesting observation here is to verify the presumption made in Section 3.3.2 about the distribution of the slacks in a chain of flights. In Section 3.3.2 we hypothesized that it is most beneficial to add the slack to the “middle” of a chain of flights in a propagation tree – under the implicit assumption that root delays are equally likely to occur in each flight. Based on the computational results, we observed that this assumption is in fact true. More specifically, wherever there is a chain of flights with “approximately the same” probabilities for root delays, the slack on the arcs closer to the beginning and the end of the chain is reduced and added to the arcs closer to the middle of the chain. However, in cases where the probabilities for the root delays are “radically different” we observe different patterns.

Finally, we observe that flight re-timing *can* have substantial improvements on the delay propagation, ranging from approximately 5 percent for the most tightly restricted instances to approximately 50 percent for the least restricted instances. Of course, there are several caveats that must accompany these results. First, we did not take into account the delay propagation associated with cabin crews or any passenger itineraries for which the second flight of the itinerary would be “held” for connecting passengers (these could easily be incorporated, given available data, and would have little impact on run times). It is not clear what impact these additions would have on the quality of the optimal solutions, as the changes would impact both the original and the modified schedules. Second, we have not taken into account recovery decisions, which again will affect propagation within both the original and the optimized schedules. Finally, we re-iterate that our surrogate objective function over-counts propagation because it considers each delay one at a time. We address this limitation through the use of a discrete-event simulation model in the next

section.

4.4.3 Goal 3 – Validity

In order to better assess the impact of the fact that the surrogate objective function does not incorporate the concurrency of delays in our optimization model, we simulated each of the schedules (the original schedule and the re-timed schedules based on the single-layer and multi-layer models) using the same probability distribution functions for the root delays as in the optimization models.

The results are summarized in Table 4.4. The first two columns describe the instance. The next three columns give the ratio of the expected amount of delay propagation (in minutes), as estimated by 2000 replications of the simulation model, divided by the value of the surrogate objective function under the original or re-timed schedule.

Data Set	Duty Restriction	Simulation / Surrogate		
		Orig. Sch.	SLM	MLM
1	0	1.187	1.224	1.180
1	5		1.261	1.200
1	10		1.275	1.203
1	15		1.288	1.202
2	0	1.168	1.193	1.166
2	5		1.224	1.170
2	10		1.238	1.170
2	15		1.252	1.158

Table 4.4: Comparing the simulation results with the surrogate objective function value.

Data Set	Duty Restrictions	Orig. Sch. Average	SLM Average	MLM Average
1	0	2692.1	2576.3 (4.3%)	2543.9 (5.5%)
1	5		2076.6 (22.8%)	2033.7 (24.4%)
1	10		1708.6 (36.5%)	1660.1 (38.3%)
1	15		1432.1 (46.8%)	1384.8 (48.6%)
2	0	2378.6	2289.0 (3.8%)	2266.7 (4.7%)
2	5		1813.9 (23.7%)	1776.4 (25.3%)
2	10		1476.6 (37.9%)	1433.0 (39.7%)
2	15		1219.3 (48.7%)	1167.8 (50.9%)

Table 4.5: Reduction in delay propagations (simulation results) – comparing different schedules.

As expected, the simulated values differ from the surrogate values. However, the ratio is fairly consistent, suggesting that the impact of ignoring concurrent delays has comparable impact on both the original and the re-timed schedules. Thus, it is not surprising that the simulated value of the re-timed schedules still demonstrates a significant improvement over the simulated value of the original schedule. Table 4.5 demonstrates this. The first two columns describe the instance. The next three columns give the expected amount of delay propagation (in minutes), as estimated by the simulation model, for each of the three schedules. Columns four and five also provide the relative improvement over the original schedule. Although these improvements are lower than the surrogate objective values, they nonetheless demonstrate a substantial opportunity for reduction in delay propagation through minor flight re-timings.

Data Set	Duty Restrictions	SLM 95% C.I.	MLM 95% C.I.
1	0	(3.2%, 5.4%)	(4.4%, 6.6%)
1	5	(21.8%, 23.9%)	(23.4%, 25.5%)
1	10	(35.6%, 37.5%)	(37.5%, 39.2%)
1	15	(46.0%, 47.6%)	(47.8%, 49.3%)
2	0	(2.6%, 5.0%)	(3.5%, 5.9%)
2	5	(22.7%, 24.8%)	(24.3%, 26.3%)
2	10	(37.0%, 38.8%)	(38.9%, 40.6%)
2	15	(47.9%, 49.5%)	(50.1%, 51.7%)

Table 4.6: 95% confidence intervals on estimated reductions in delay propagation.

Based on simulation results, we also constructed 95% confidence intervals on the reductions in delay propagations. We used common random numbers in order to decrease the estimation error and used paired-t statistic to find the values provided in Table 4.6.

4.4.4 Discussion

We conclude this section with a few final observations.

First, we note that changing flight times can make key passenger itineraries infeasible by decreasing the connection time between them beyond that which is feasible for a passenger to connect. Such itineraries can be protected by adding constraints of the form:

$$(4.22) \quad y_{f_1, f_2} = s_{f_1, f_2} - x_{f_1} + x_{f_2}$$

$$(4.23) \quad y_{f_1, f_2} \geq 0$$

which says that, for a protected itinerary (f_1, f_2) , the time between the scheduled arrival of flight f_1 and the scheduled departure of flight f_2 must not decrease below the minimum passenger connection time. Observe, however, that we do not include delay propagation from flight f_1 to flight f_2 in the model, because we do not assume that flights are delayed to await connecting passengers.

Second, we observe that – as with any model – the re-timing solutions that result from our models will only be starting points, which will need to be fine-tuned before implementation. In particular, although the 0-minute model may be overly restrictive in terms of maintaining the current crew costs and feasibility, the 15-minute model may result in some crew violations. These violations would have to be corrected manually, but we suggest that substantial benefits can still remain even after these modifications.

Third, to partially reduce this post-processing, we suggest that one way to reduce crew infeasibilities would be to identify the largest amount by which the length of the duty could increase without violating elapsed time limits and/or changing cost the dominant cost from the flight time to the elapsed time. Then, for each duty, a constraint of form:

$$(4.24) \quad -x_{f'} + x_{f''} \leq e^{max} - e$$

could be used to ensure that the length of the duty not increase beyond this limit. In the above constraint, f' and f'' are the first and the last flights of a duty respectively, e is the elapsed time of that particular duty, and e^{max} is the maximum elapsed time in a duty.

Finally, we can consider a hypothetical case (for example, see Figure 4.5) where the tree structure is violated. In this particular example, the root flight, f_0 is connected to two flights, f_2 and f_1 due to the sharing of the cockpit crew and the aircraft, respectively. Subsequently, flight f_2 shares the same cockpit crew with flight f_3 while the same aircraft that is assigned to f_1 , flies f_3 as well. In this case, delay propagations form a *directed acyclic graph* (DAG) instead of a tree. Therefore, each node in the corresponding worst-case “propagation DAG” may have more than one parent.

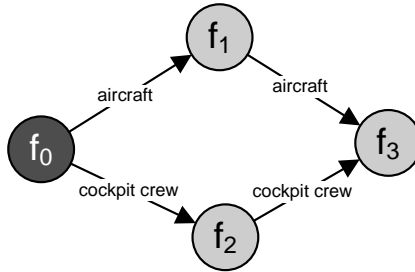


Figure 4.5: An example of a directed acyclic graph. In this case, flight f_3 has more than one parent.

Under such assumption, we can define the following notation:

$\mathcal{D}_{f_0}^m$ set of flights in the worst-case propagation DAG associated with root flight f_0 and root delay m (excluding f_0)

- $\mathcal{R}_{f_0}^m(f)$ the set of parent nodes associated with flight f in $\mathcal{D}_{f_0}^m$
 $r_{f_0}^m(f)$ an element of the set of parent nodes associated with flight f in $\mathcal{D}_{f_0}^m$

Consequently, we need to substitute the constraints (4.12) and (4.13) in MLM with:

$$(4.25) \quad d_{f_0,f}^m \geq m - y_{f_0,f} \quad \forall (f_0 \in \mathcal{F}, f \in \mathcal{D}_{f_0}^m : r_{f_0}^m(f) = f_0), \forall m \in \mathcal{M}, \forall r_{f_0}^m(f) \in \mathcal{R}_{f_0}^m$$

$$(4.26) \quad d_{f_0,f}^m \geq d_{f_0,r_{f_0}^m(f)}^m - y_{r_{f_0}^m(f),f} \quad \forall (f_0 \in \mathcal{F}, f \in \mathcal{D}_{f_0}^m : r_{f_0}^m(f) \neq f_0), \forall m \in \mathcal{M}, \forall r_{f_0}^m(f) \in \mathcal{R}_{f_0}^m$$

Here we should note that we did not observe such case in the data sets mentioned in Section 4.4.1.

4.5 Conclusions and Future Research

Airline delays have significant negative impact on airline costs, passenger convenience and productivity, and the environment. One major cause of delay is the down-stream propagation of initial delays to subsequent flights. This issue is exacerbated by the fact that slack is undesirable from a planning perspective, as it “wastes” costly resources, but is critical from an operational perspective, as it can be used to absorb disruptions and prevent their propagation.

Addressing operational concerns in the planning process can be quite challenging, however. First, metrics for evaluating the operational performance of a planned schedule must be developed. Second, cost functions must be developed to trade-off planned and (anticipated) operational costs. Finally, these cost functions must be incorporated in an already-challenging planning process. In particular, because delay propagation spans across multiple resources, schedule design, fleet, crew scheduling, and maintenance routing must all be considered concurrently.

As an intermediate measure to partially decrease the propagation of delay while

these other challenging topics are studied by the research community, we propose to modify flight departure times within the framework of an existing airline plan. By reallocating the existing slack to those flight connections that are most prone to delay propagation, we can reduce downstream impacts without changing planned crew or fleet costs and without changing revenue projections. Our computational results show significant opportunities for improvement without any increase in planned costs.

Future research in this area of course includes the three issues raised above – metrics for evaluating the robustness of a planned schedule, cost functions for computing the trade-off between planned costs and anticipated operational costs, and methods for incorporating these cost functions in the planning process. In the shorter time horizon, our research could be expanded by more explicitly addressing correlations between different root delays. Finally, we are interested in including recovery decisions (e.g. canceling flights, swapping aircraft, and calling in reserve crews) in our model to capture their impacts on the operational performance of a planned schedule under disruptions.

CHAPTER V

An Integer Programming Approach to Generating Airline Crew Pairings

5.1 Introduction

Generating *crew pairings* – sequences of flights that can be flown by a single crew – is a critical task not only for solving the airline crew scheduling problem, but for the broader problems of *integrated airline planning*, *robust planning*, and *automated recovery* as well. A significant challenge that must be overcome when solving these problems is the large number of potential pairings – a reasonably-sized flight network can have on the order of millions or even billions of feasible pairings. Candidate pairings are therefore typically identified via *column generation*, rather than explicitly enumerated (see Lavoie et al [35]).

Column generation leads to additional challenges, however, due to the non-linear cost function and complex feasibility rules associated with assigning crews to flight sequences. As a result, there has been a broad body of research devoted to methods for generating promising crew pairings. This research (summarized in Section 5.2) has typically focused on the development of highly-customized algorithms for solving the airline crew scheduling problem, and several commercial software packages have successfully implemented these algorithms. On the other hand, these customized *crew pairing generators* require significant implementation efforts. This serves as a

major impediment to researchers seeking to solve larger and more challenging problems (such as integrated planning, robust planning, and automated recovery) for which crew pairing generation is only one small part. The efforts required to implement customized crew pairing algorithms and embed them in a broader framework (for example, in a model that integrates crew scheduling with other stages of the planning process) greatly reduce the number of new ideas that researchers can test and develop, serving as a significant barrier-to-entry in this field of research.

On the other hand, an integer programming approach to generating crew pairings can greatly facilitate the prototyping and evaluation of new research efforts, by enabling the use of traditional solution techniques such as branch-and-bound which are commonly found in off-the-shelf commercial solvers. Although the non-linear cost function and feasibility rules of crew pairings pose challenges for integer programming, we present a novel modeling approach, based on both *connection variables* and *marker variables*, that overcomes these obstacles. The resulting model can be solved using standard IP techniques and solvers, and thus can more easily be embedded in a larger problem framework.

This provides two benefits to researchers by enabling them to more quickly prototype new ideas. First, the potential value to be provided by a new approach can be evaluated. For example, test instances might be solved to estimate the potential savings associated with integrating two problems, relative to solving them sequentially. Second, if the idea shows promise, then the computational performance of the proposed model and/or algorithm can be evaluated, using the run times associated with using the IP-based crew pairing generator as an upper bound. Of course, a more sophisticated (but challenging-to-implement) mechanism for generating pairings could then be used to further improve run time. By benchmarking the run time

of the IP-based pairing generator against other pairing generators, more accurate estimates of the potential performance could be achieved.

The contributions of this research are thus two-fold. First, we present a new way of modeling complex non-linear constraints and cost functions in a scheduling context. Second, we show how this approach enables us to develop a crew pairing generator that can facilitate research in an important and emerging research area – that of integrated airline planning, robust planning, and automated recovery.

The outline of this chapter is as follows. In Section 5.2 we describe the crew pairing problem and review the related literature. Our approach is presented in Section 5.3, followed by computational results in Section 5.4. Finally, conclusions and future research are discussed in Section 5.6.

5.2 The Crew Pairing Problem

In this section we formally present the crew pairing problem and review the related literature.

5.2.1 Problem Statement

The objective of the *airline crew scheduling* problem is to find a least-cost assignment of crews (cockpit or cabin) to flights. This is typically done in two major steps. The first step, the *crew pairing* (CP) problem, partitions the flights into pairings. Three separate versions of this problem are typically solved – one for the flights which are flown each day, one for the flights that are only flown a few times per week, and one for the transition between the end of one scheduling period and the start of the next. The second step, known as either the *crew assignment* problem or *rostering* problem, assigns specific crews to these pairings. We refer the reader to Barnhart et al [5] for a more detailed discussion of the crew scheduling problem while we focus

on the crew pairing problem.

A pairing is a multi-day sequence of flights that begins and ends at a crew base and can feasibly be flown by a single crew. Pairings are made up of individual *duties* (one-day work schedules), which are interspersed with periods of rest. The duties are in turn sequences of individual flights with brief periods between them to allow for connecting between flights.

Pairings and duties are both subject to a lengthy list of feasibility rules, stemming from government regulations as well as labor negotiations. These include:

- The total flying time within a duty period must not exceed an upper bound; the total elapsed time is limited as well.
- There is a minimum and maximum amount of time permitted between two consecutive flights within a duty.
- There is a minimum amount of time permitted between two consecutive duties within a pairing.
- There is a maximum number of duties permitted within a pairing.

In addition to the complexity caused by enforcing these feasibility rules, pairings also suffer from a non-linear pay structure. The cost of a pairing is the maximum of two terms – the sum of the costs of the individual duties, and a fixed percentage of the elapsed time of the pairing (also known as time away from base, or *TAFB*). The cost corresponding to an individual duty period is itself the maximum of three terms – the total flying time in the duty, a fixed percentage of the elapsed time of

the duty, and a minimum guaranteed payment per duty. These can be written as :

$$(5.1) \quad c_d = \max\{fly_d, \beta e_d, \gamma\}$$

$$(5.2) \quad c_p = \max\left\{\left(\sum_d c_d\right), \alpha T_p\right\}$$

where:

c_d = cost of duty d

fly_d = total flying time of duty d

β = percentage of elapsed time of a duty applied to duty cost function

$(0 < \beta < 1)$

e_d = elapsed time of duty d

γ = minimum guaranteed duty payment

c_p = cost of pairing p

α = percentage of elapsed time of a pairing applied to pairing cost function $(0 < \alpha < 1)$

T_p = elapsed time of pairing p .

5.2.2 Formulation

Because of its non-linear cost function and complicating feasibility rules, the crew pairing problem is commonly modeled as a set partitioning (SP) problem (see Balas and Padberg [4] for a broad survey of the set partitioning problem and Ryan and Falkner [47] for a discussion of the properties of SP models as applied specifically to scheduling problems). Although approaches other than SP (for example, Hoffman and Padberg [21] and Yan and Tu [57]) have sometimes been successful in solving the crew pairing problem, especially on small instances or under less complicated regulations, SP -based approaches are typically used because they enable the embedding

of many complex rules within the variable definition.

The set-partitioning formulation of the crew pairing problem is:

$$(5.3) \quad \min \sum_{p \in \mathcal{P}} c_p x_p$$

subject to

$$(5.4) \quad \sum_{p \in \mathcal{P}} \delta_{fp} x_p = 1 \quad \forall f \in \mathcal{F}$$

$$(5.5) \quad x_p \in \{0, 1\} \quad \forall p \in \mathcal{P}$$

Where:

\mathcal{F} = set of all flights

\mathcal{P} = set of all feasible pairings

c_p = cost of pairing p

$\delta_{fp} = 1$ if flight f is included in pairing p and 0 otherwise.

and:

$$x_p = \begin{cases} 1 & \text{if pairing } p \text{ is chosen} \\ 0 & \text{otherwise} \end{cases} \quad \forall p \in \mathcal{P}$$

The objective (5.3) minimizes the cost of the chosen pairings. The cover constraints (5.4) and integrality constraints (5.5) require that for each flight f , exactly one pairing is chosen containing that flight.

5.2.3 Solution Techniques

Although the SP formulation captures the complex feasibility rules and non-linear cost function of the crew pairing problem implicitly in the variable definitions, it results in an exponential number of decision variables, making it intractable to solve explicitly for all but the smallest of flight networks. Instead, it is typically solved using *branch-and-price* (Barnhart et al [5]). In this approach, a large-scale integer program is solved with the *branch-and-bound* algorithm, but with the individual linear programs solved using *column generation*. Column generation was originally used by Gilmore and Gomory [20] to solve large-scale linear programming problems such as the cutting stock problem. In this approach, first a minimal subset of variables is chosen and a *restricted master (RM)* problem is constructed from them. *RM* is solved to optimality and then an optimization problem called the *sub-problem* or the *pricing problem* uses the resulting dual information to find a new non-basic variable with negative reduced cost. This variable is pivoted into *RM* and the process repeats. Such an approach eliminates the need to explicitly enumerate all of the variables (in this case, pairings).

When using branch-and-price to solve the crew pairing problem, the sub-problem is to find a feasible pairing with negative reduced cost. However, within this sub-problem, all of the complex feasibility rules, as well as the non-linear cost function, must be addressed. Therefore much of the crew scheduling literature is dedicated to finding efficient methods to solve this sub-problem, which is also known as *pairing generation*. The most common approaches to solving the pricing problem are *enumeration-based methods*, *heuristic methods*, and the *multi-label-shortest-path (MLSP)* algorithm.

Enumeration-based approaches, in which the variables are not all explicitly stored

in the restricted master but *are* all enumerated iteratively within the pricing problem, have been successfully used in some set partitioning problems (see Garfinkel and Nemhauser [19] and Marsten [40]). The benefits of such an approach are primarily in memory-management, as the full set of feasible variables must still be visited. Clearly, the success of such approaches is greatly dependent on the size of the problem instance, and is therefore problematic for solving crew scheduling across a large set of flights. Nonetheless, enumeration-based approaches have seen some limited success in the crew pairing problem with the addition of some basic forms of bounding as is seen in Klabjan and Schwan [30] and Makri and Klabjan [39].

Randomized heuristics have also been used in order to generate promising pairings. Such approaches do not guarantee an optimal solution to the pricing problem (i.e., the variable with the most negative reduced cost), and therefore cannot provide a certificate of optimality for *RM*. Nonetheless, these approaches can be useful in generating promising columns and, in particular, can be successfully mixed with exact techniques to form an efficient hybrid algorithm (see Klabjan et al [29]). Another heuristic approach that has been successful in generating crew pairings is *constraint programming*, as is seen in Fahle et al [17]. Again, such an approach can also be combined with exact approaches.

The *multi-label shortest-path algorithm*, originally adapted to the crew pairing problem by Desrochers and Soumis [15] and Lavoie et al [35], is perhaps the most well-known approach to generating crew pairings. In this algorithm, each pairing corresponds to a path in a *flight network*, with the cost of each path defined to be the cost of the corresponding pairing minus the sum of the dual values associated with the flights in that path. A collection of labels is used to capture the feasibility criteria and to compute the path cost. *Dominance* is used to avoid explicit

enumeration by pruning portions of the dynamic programming tree. However, in this specific application, the notion of pruning by dominance may also rule out the optimal solution.

The fact that flights cannot be repeated within a pairing can lead to invalid pruning in the multi-label shortest path algorithm, because no one pairing fragment can fully dominate another. Here, we provide a simplistic example to demonstrate how this can happen. Consider the set of eight flights presented in Table 5.1.

Flight	Origin	Destination	Departure	Arrival	Fly Time
A	1	2	9	11	2
B	2	3	13	15	2
C	3	2	6	12	6
D	2	4	16	18	2
E	4	1	10	12	2
F	1	2	13	15	2
G	2	3	14	17	3
H	3	1	22	26	4

Table 5.1: A counter example for dominance in MLSP

Assume that the minimum guaranteed duty pay (γ) is four, the fraction of the elapsed duty time (β) is 0.6, the maximum flying time in a duty (φ) is eight, the maximum duty elapsed time (δ) is twelve, the maximum number of duties (D) is three, the minimum turn time (τ) is 0.5, and the only crew base is station 1.

This problem instance has four feasible solutions:

- Pairings $\{AG|CB|H\}$ and $\{FD|E\}$, with cost 25 (where $|$ delineates two duties)
 - Pairings $\{AB|C|GH\}$ and $\{FD|E\}$, with cost 25.2
 - Pairings $\{AB|CD|E\}$ and $\{F|GH\}$, with cost 27.2
- and
- Pairings $\{AG|CD|E\}$ and $\{F|B|H\}$, with cost 29.

The first of these solutions is uniquely optimal and requires pairing $\{AG|CB|H\}$. Suppose that the restricted master at some intermediate stage contains *all* feasible pairings except this one. This pairing should be identified by the pricing problem as having negative reduced cost and then be pivoted into RM . However, given the dual values for the current solution to RM , the fragment $\{AB|C\dots\}$ dominates the fragment $\{AG|C\dots\}$ – it has lower cost, lower dual component, less flying time, and the same start/end flights. Thus, the fragment $\{AG|C\dots\}$ would be pruned in $MLSP$. However, the fragment $\{AB|C\dots\}$ does not lead to a negative reduced cost pairing (because it cannot repeat flight B), whereas $\{AG|C\dots\}$ would. Thus, $MLSP$ will return a false certificate of optimality.

5.3 An Integer Programming Approach to Generating Pairings

5.3.1 Motivation

All of the solution techniques outlined in Section 5.2.3 require the implementation of sophisticated, customized data structures and algorithms. Although these implementations exist within commercial solvers used by the airlines, they are either unavailable to academic researchers, or when they are available, they often require substantial effort to understand, modify, and integrate into the framework of a broader research question.

We therefore present a novel approach, in which the crew pairing generation problem is defined as an integer program within a network flow framework. A feasible solution, i.e., feasible pairing, is a path or a sequence of flights and the objective is to find the most negative reduced-cost pairing. This approach can be implemented without any sophisticated computer coding required. Moreover, as IP commercial solvers become faster and more capable, these benefits are gained without additional work by the researchers. The user can also exploit several possible environmental

settings in commercial solvers (e.g. optimality tolerance, branching strategies, number of feasible solutions found, hot starts, etc.) in order to obtain desirable solution quality in appropriate time.

5.3.2 Mathematical Model

The proposed mixed-integer programming (MIP) model is based on a set of binary decision variables (*connection variables*) that determine whether two flights immediately follow each other. These variables also capture the added information of whether the two flights follow each other in the same duty or span the overnight between two duties. A second set of *marker variables* indicating placement information such as flights which begin and end the pairing. A solution to this MIP is a sequence of flights that forms a feasible pairing. Balance constraints guarantee that the connections form a path while side constraints and auxiliary variables capture the cost structure and feasibility rules. The objective function minimizes the reduced cost of the path, which is the true cost of the pairing minus the sum of the dual values of the flights that appear in that pairing. The mathematical model is presented here for the daily problem, in which each flight is assumed to repeat daily. We revise the model for the weekly and transition problems in Section 5.3.3.

1. Parameters Defined Over the Set of Flights (\mathcal{F})

Note: All time units are in hours.

$\pi_f \in \mathbb{R}^+$	$\forall f \in \mathcal{F}$	dual value of flight f
$t_f \in [0, 24]$	$\forall f \in \mathcal{F}$	departure (takeoff) time of flight f
$l_f \in [0, 24]$	$\forall f \in \mathcal{F}$	arrival (landing) time of flight f
$b_f > 0$	$\forall f \in \mathcal{F}$	total flying (block) time of flight f
o_f	$\forall f \in \mathcal{F}$	origin of flight f

d_f $\forall f \in \mathcal{F}$ destination of flight f

2. Parameters Corresponding to Regulations

$D \in \mathbb{Z}^+$ maximum number of duties in the pairing

$\alpha \in (0, 1)$ percentage of elapsed time of a pairing applied to the pairing
cost function

$\beta \in (0, 1)$ percentage of elapsed time of a duty applied to the duty cost
function

ω home location or crew base (airport)

$\tau \in \mathbb{R}^+$ minimum required time between two consecutive flights in a duty

$\sigma \in \mathbb{R}^+$ maximum permitted time between two consecutive flights in a duty

$\gamma \in \mathbb{R}^+$ minimum guaranteed duty payment

$\varphi \in \mathbb{R}^+$ maximum duty flying time

$\delta \in \mathbb{R}^+$ maximum duty elapsed time

$\rho \in \mathbb{R}^+$ minimum rest time between consecutive duties

Note: $\gamma < \varphi < \delta$

3. Sets

$\mathcal{D} = \{1, \dots, D\}$ the set of all possible duty indices
(i.e., first duty, second duty, etc.)

$\mathcal{D}' = \{1, \dots, D - 1\}$ the set of all possible duty indices except the last

$\mathcal{D}'' = \{2, \dots, D\}$ the set of all possible duty indices except the first

$\mathcal{D}''' = \{2, \dots, D - 1\}$ the set of all possible duty indices except the first and last

$$\mathcal{S} = \{(f_1, f_2) \in \mathcal{F} \text{ s.t. } d_{f_1} = o_{f_2}, b_{f_1} + b_{f_2} \leq \varphi \text{ and}$$

$$\left. \left\{ \begin{array}{l} t_{f_2} - l_{f_1} \geq \tau \\ t_{f_2} - l_{f_1} \leq \sigma \\ l_{f_2} - t_{f_1} \leq \delta \end{array} \right\} \text{ OR } \left\{ \begin{array}{l} l_{f_1} > t_{f_2} \\ t_{f_2} + 24 - l_{f_1} \geq \tau \\ t_{f_2} + 24 - l_{f_1} \leq \sigma \\ l_{f_2} + 24 - t_{f_1} \leq \delta \end{array} \right\} \right\}$$

$$\mathcal{O} = \{(f_1, f_2) \in \mathcal{F} \text{ s.t. } d_{f_1} = o_{f_2}\}$$

$$\mathcal{B} = \{f \in \mathcal{F} \text{ s.t. } o_{f_1} = \omega\}$$

$$\mathcal{E} = \{f \in \mathcal{F} \text{ s.t. } d_{f_1} = \omega\}$$

In other words:

- \mathcal{S} : Set of all flight pairs that can be sequential in the same duty. The destination of the first flight must match the origin of the second, the total flying time of the two cannot exceed the total permitted flying time of a duty, the time interval between the first flight's arrival and the second flight's departure must be within the minimum required and maximum permitted times, and the total elapsed time spanned by the two flights cannot exceed the elapsed time limit of a duty. Note that when the two flights "cross midnight" (e.g. the first flight has an 11 p.m. arrival and the second flight has a 1 a.m. departure), then we need to shift the takeoff of the second flight 24 hours to recognize that it is the subsequent day's flight.
- \mathcal{O} : Set of all flight pairs that can "span" a duty, i.e., one flight ends a duty and the second flight starts the next duty (after an adequate period of rest). This set includes all flight pairs where the destination of the first matches the origin of the second – if the time between them is inadequate for the rest period, then the second flight can be shifted a day later and the rest

period increased by 24 hours.

- \mathcal{B} : Set of all flights that originate at the crew base and thus can begin the pairing.
- \mathcal{E} : Set of all flights that terminate at the crew base and thus can end the pairing.

4. Other Parameters

The following parameters capture the overnight rest (r) and sit (s) times between two consecutive flights.

$$r_{(f_1, f_2) \in \mathcal{F}} = \begin{cases} t_{f_2} - l_{f_1} & \text{if } t_{f_2} - l_{f_1} \geq \rho \\ t_{f_2} + 24 - l_{f_1} & \text{if } t_{f_2} > l_{f_1} \text{ and } t_{f_2} - l_{f_1} < \rho \\ t_{f_2} + 24 - l_{f_1} & \text{if } l_{f_1} > t_{f_2} \text{ and } t_{f_2} + 24 - l_{f_1} \geq \rho \\ t_{f_2} + 48 - l_{f_1} & \text{if } l_{f_1} > t_{f_2} \text{ and } t_{f_2} + 24 - l_{f_1} < \rho \end{cases}$$

$$s_{(f_1, f_2) \in \mathcal{S}} = \begin{cases} t_{f_2} - l_{f_1} & \text{if } t_{f_2} > l_{f_1} \\ t_{f_2} + 24 - l_{f_1} & \text{if } t_{f_2} < l_{f_1} \end{cases}$$

Where:

- r_{f_1, f_2} is the rest period between two flights that follow each other in two consecutive duties. This is the departure time of the second flight minus the arrival time of the first flight. If this rest period is less than the minimum then the second flight must be shifted a day later and the rest period increased by 24 hours. If the arrival of the first flight is later than the departure of the second flight, then the second flight must also be shifted by a day. Finally, it may be necessary to shift a flight two days both to account for time of day (i.e., second flight precedes the first) and if the length of

the rest period resulting from the initial shift of one day is less than the minimum rest requirement.

- s_{f_1, f_2} is the time between two flights that follow each other in the same duty (*sit time*), where 24 hours must be added to the second flight if it must be shifted a day later.

5. Decision Variables

$v_f \in \{0, 1\}$	$\forall f \in \mathcal{B}$	whether f is the first flight of the pairing
$w_{f,d} \in \{0, 1\}$	$\forall f \in \mathcal{E}, d \in \mathcal{D}$	whether f is the last flight of the pairing and the pairing is d duties in length
$x_{f_1, f_2, d} \in \{0, 1\}$	$\forall (f_1, f_2) \in \mathcal{S}, d \in \mathcal{D}$	whether f_1 is followed by f_2 within the d^{th} duty
$y_{f_1, f_2, d} \in \{0, 1\}$	$\forall (f_1, f_2) \in \mathcal{O}, d \in \mathcal{D}'$	whether f_1 ends the d^{th} duty and is followed by f_2 starting the $d + 1^{\text{st}}$ duty
$z_{f,d} \in \{0, 1\}$	$\forall f \in \mathcal{F}, d \in \mathcal{D}$	whether f appears in the d^{th} duty
$u_d \in \{0, 1\}$	$\forall d \in \mathcal{D}$	whether the pairing contains at least d duties
$C \in \mathbb{R}^+$		cost of the pairing
$T \in \mathbb{R}^+$		elapsed time of the pairing (TAFB)
$c_d \in \mathbb{R}^+$	$d \in \mathcal{D}$	cost of the d^{th} duty
$e_d \in \mathbb{R}^+$	$d \in \mathcal{D}$	elapsed time of the d^{th} duty

6. Objective Function

$$(5.6) \quad \min C - \sum_{f \in \mathcal{F}} \sum_{d \in \mathcal{D}} \pi_f z_{f,d}$$

The objective function (5.6) minimizes the reduced cost of the pairing – the true cost of the pairing minus the sum of the dual contributions of each flight included in the pairing.

7. Constraints

(a) Cost Constraints

$$(5.7) \quad c_d \geq \gamma u_d \quad \forall d \in \mathcal{D}$$

$$(5.8) \quad c_d \geq \sum_{f \in \mathcal{F}} b_f z_{f,d} \quad \forall d \in \mathcal{D}$$

$$(5.9) \quad c_d \geq \beta e_d \quad \forall d \in \mathcal{D}$$

$$(5.10) \quad T = \sum_{d \in \mathcal{D}} e_d + \sum_{(f_1, f_2) \in \mathcal{O}} \sum_{d \in \mathcal{D}'} r_{f_1, f_2} y_{f_1, f_2, d}$$

$$(5.11) \quad C \geq \sum_{d \in \mathcal{D}} c_d$$

$$(5.12) \quad C \geq \alpha T$$

Constraints (5.7) through (5.12) are used to compute the cost of the pairing, as defined in Section 5.2.1. Constraints (5.7), (5.8), and (5.9) define the cost of each duty as the maximum of: the minimum guaranteed payment, the total flying time of the duty, and a fraction of the total elapsed time of the duty. [Note that if the pairing contains fewer than D duties, the right-hand side of these constraints will be zero for the appropriate values of d]. Constraints (5.11) and (5.12) define the cost of the pairing as the maximum of the total cost of the duties in the pairing and a fraction of the total elapsed time of the pairing, which is defined in constraint (5.10) as the sum of the elapsed time of the duties in the pairing plus the sum of the

rest periods between duties.

(b) **Constraints Defining Auxiliary Variables**

Constraints (5.13) through (5.20) enforce the relationship between the auxiliary variables, u and z , and the main decision variables (x , y , v and w).

$$(5.13) \quad u_1 = 1$$

$$(5.14) \quad u_d = \sum_{(f_1, f_2) \in \mathcal{O}} y_{f_1, f_2, d-1} \quad \forall d \in \mathcal{D}''$$

$$(5.15) \quad z_{f,1} = \sum_{(f_1, f) \in \mathcal{S}} x_{f_1, f, 1} \quad \forall f \in \mathcal{F} \setminus \mathcal{B}$$

$$(5.16) \quad z_{f,1} = v_f + \sum_{(f_1, f) \in \mathcal{S}} x_{f_1, f, 1} \quad \forall f \in \mathcal{B}$$

$$(5.17) \quad z_{f,d} = \sum_{(f, f_1) \in \mathcal{O}} y_{f, f_1, d} + \sum_{(f, f_1) \in \mathcal{S}} x_{f, f_1, d} \quad \forall f \in \mathcal{F} \setminus \mathcal{E}, \quad \forall d \in \mathcal{D}'''$$

$$(5.18) \quad z_{f,d} = w_{f,d} + \sum_{(f, f_1) \in \mathcal{O}} y_{f, f_1, d} + \sum_{(f, f_1) \in \mathcal{S}} x_{f, f_1, d} \quad \forall f \in \mathcal{E}, \quad \forall d \in \mathcal{D}'''$$

$$(5.19) \quad z_{f,D} = \sum_{(f, f_1) \in \mathcal{S}} x_{f, f_1, D} \quad \forall f \in \mathcal{F} \setminus \mathcal{E}$$

$$(5.20) \quad z_{f,D} = w_{f,D} + \sum_{(f, f_1) \in \mathcal{S}} x_{f, f_1, D} \quad \forall f \in \mathcal{E}$$

Constraint (5.13) ensures that the pairing contains at least one duty. For all $d > 1$, constraint (5.14) states that the d^{th} duty exists if it contains an overnight connection from the preceding duty. Constraints (5.15) state that a flight exists in the first duty if it is preceded by another flight in that duty; if the flight originates at the crew base, then it can also be the first flight of the pairing (constraints (5.16)). If a flight appears in any duty other than the first, then it must either be the last flight of that duty (and thus precede a rest period) or be followed by another flight in that duty (5.17). If the flight ends at the crew base then it may also be the last flight

in the pairing (constraints (5.18)). Constraints (5.19) and (5.20) address the special case where there can be no subsequent duty and thus a flight cannot precede a rest period.

(c) **Duty Constraints**

$$(5.21) \quad \sum_{f \in \mathcal{F}} b_f z_{f,d} \leq \varphi \quad \forall d \in \mathcal{D}$$

$$(5.22) \quad e_d = \sum_{f \in \mathcal{F}} b_f z_{f,d} + \sum_{(f,f_1) \in \mathcal{S}} s_{f,f_1} x_{f,f_1,d} \quad \forall d \in \mathcal{D}$$

$$(5.23) \quad e_d \leq \delta \quad \forall d \in \mathcal{D}$$

Constraints (5.21) through (5.23) guaranty the feasibility of the duties. Constraint (5.21) ensures that the total flying time in the d^{th} duty does not exceed the maximum permitted flying time. Constraint (5.22) defines the elapsed time of the d^{th} duty as the sum of all flying times in the duty plus the sum of all sit times in it. Constraint (5.23) prevents the d^{th} duty from violating the maximum elapsed time.

(d) **Pairing Constraint**

$$(5.24) \quad \sum_{d \in \mathcal{D}} z_{f,d} \leq 1 \quad \forall f \in \mathcal{F}$$

Constraint set (5.24) states that each flight f can appear at most once in the pairing.

(e) **Balance Constraints**

Constraints (5.25) through (5.32) are analogous to “balance constraints” in a network flow problem, forcing the chosen flights to form a sequential path.

$$(5.25) \quad \sum_{f \in \mathcal{F}} v_f = 1$$

$$(5.26) \quad \sum_{f_1 \in \mathcal{F}, (f_1, f) \in \mathcal{S}} x_{f_1, f, 1} = \sum_{f_1 \in \mathcal{F}, (f, f_1) \in \mathcal{S}} x_{f, f_1, 1} + \sum_{f_1 \in \mathcal{F}, (f, f_1) \in \mathcal{O}} y_{f, f_1, 1} \quad \forall f \in \mathcal{F} \setminus \mathcal{B} \cup \mathcal{E}$$

$$(5.27) \quad v_f + \sum_{f_1 \in \mathcal{F}, (f_1, f) \in \mathcal{S}} x_{f_1, f, 1} = \sum_{f_1 \in \mathcal{F}, (f, f_1) \in \mathcal{S}} x_{f, f_1, 1} + \sum_{f_1 \in \mathcal{F}, (f, f_1) \in \mathcal{O}} y_{f, f_1, 1} \quad \forall f \in \mathcal{B}$$

$$(5.28) \quad \sum_{f_1 \in \mathcal{F}, (f_1, f) \in \mathcal{S}} x_{f_1, f, 1} = \sum_{f_1 \in \mathcal{F}, (f, f_1) \in \mathcal{S}} x_{f, f_1, 1} + \sum_{f_1 \in \mathcal{F}, (f, f_1) \in \mathcal{O}} y_{f, f_1, 1} + w_{f, 1} \quad \forall f \in \mathcal{E}$$

$$(5.29) \quad \sum_{f_1 \in \mathcal{F}, (f_1, f) \in \mathcal{S}} x_{f_1, f, d} + \sum_{f_1 \in \mathcal{F}, (f_1, f) \in \mathcal{O}} y_{f_1, f, d-1} = \sum_{f_1 \in \mathcal{F}, (f, f_1) \in \mathcal{S}} x_{f, f_1, d} \\ + \sum_{f_1 \in \mathcal{F}, (f, f_1) \in \mathcal{O}} y_{f, f_1, d} \quad \forall f \in \mathcal{F} \setminus \mathcal{E}, \forall d \in \mathcal{D}'''$$

$$(5.30) \quad \sum_{f_1 \in \mathcal{F}, (f_1, f) \in \mathcal{S}} x_{f_1, f, d} + \sum_{f_1 \in \mathcal{F}, (f_1, f) \in \mathcal{O}} y_{f_1, f, d-1} = \sum_{f_1 \in \mathcal{F}, (f, f_1) \in \mathcal{S}} x_{f, f_1, d} \\ + \sum_{f_1 \in \mathcal{F}, (f, f_1) \in \mathcal{O}} y_{f, f_1, d} + w_{f, d} \quad \forall f \in \mathcal{E}, \forall d \in \mathcal{D}'''$$

$$(5.31) \quad \sum_{f_1 \in \mathcal{F}, (f_1, f) \in \mathcal{S}} x_{f_1, f, D} + \sum_{f_1 \in \mathcal{F}, (f_1, f) \in \mathcal{O}} y_{f_1, f, D-1} = \sum_{f_1 \in \mathcal{F}, (f, f_1) \in \mathcal{S}} x_{f, f_1, D} + w_{f, D} \quad \forall f \in \mathcal{E}$$

$$(5.32) \quad \sum_{f_1 \in \mathcal{F}, (f_1, f) \in \mathcal{S}} x_{f_1, f, D} + \sum_{f_1 \in \mathcal{F}, (f_1, f) \in \mathcal{O}} y_{f_1, f, D-1} = \sum_{f_1 \in \mathcal{F}, (f, f_1) \in \mathcal{S}} x_{f, f_1, D} \quad \forall f \in \mathcal{E}$$

Constraint (5.25) ensures the existence of a first flight for the pairing. Constraints (5.26) state that if a flight appears in the first duty (and hence is preceded by a flight in the first duty), then it must either be followed by another flight in the first duty or precede a rest period leading into the second duty. If the flight originates at the crew base, then it can also be the first flight of the pairing (and hence the additional term in the left-hand side of constraints (5.27)). If the flight terminates at the crew base, then it can also be the last flight of the pairing (leading to the additional term in the right-hand side of constraints(5.28)).

For $d > 1$, a flight appears in the d^{th} duty if it is either preceded by another flight in the duty or follows a rest period. If a flight appears in the d^{th} duty, then it must be followed by a flight within the d^{th} duty or precede a rest period leading into the $d + 1^{\text{st}}$ duty (constraints (5.29)). If the flight

terminates at the crew base, then it may also end the pairing (and hence the additional term in the right-hand side of constraint (5.30)).

Finally, if the flight appears in the D^{th} duty, then it cannot precede a rest period, as there can be no subsequent duties (and hence the removal of the corresponding term from constraints (5.31) and (5.32)).

5.3.3 Discussion

In this section we make some brief observations about this model.

1. Weekly and Transition Problems

In Section 5.3.2, we formulated the integer programming-based approach for the crew pairing problem, focusing on the daily version. In the daily problem, each flight is assumed to operate every day of the week. Each pairing in turn is flown by a different crew starting each day of the week. Thus, a flight cannot appear more than once in a pairing. Furthermore, a flight on one day can connect to another flight the following day, due to the assumption that every flight is flown every day. In the weekly problem, we consider flights that do not occur every day of the week, as well as those flights from broken pairings – pairings which include a flight that is not actually part of the schedule. In this case the model remains the same. What changes is the parameters of this model – specifically, the set \mathcal{S} of flight connections. In this case, it is necessary to think of flights as corresponding to not only locations and time of day, but also to a specific day of week. The connections in set \mathcal{S} must accommodate these days of week. For example, a Tuesday flight can connect to a Wednesday flight and thus this connection will appear in \mathcal{S} , but this flight cannot connect to a Monday version of the same flight. This approach is also valid for the transition problem,

where flights at the end of one schedule must be combined with flights from the beginning of another schedule to create pairings. Finally, it is not uncommon in the weekly and transition problems for deadheads to be used. These are flights where the crew flies as a passenger solely for the purpose of repositioning. In order to accommodate this, deadhead flight opportunities would be included in the input flight set for the model (see [6] for a discussion of how to choose promising deadhead flights). Furthermore, these deadhead flights would not necessarily be restricted to a single use (i.e., constraint set (5.24) would not be applied).

2. Reusability

Recall that crew pairings will typically be generated many times when solving the crew pairing problem – not just to find an optimal solution to the LP relaxation, but then again at each node of the branch-and-bound tree. One advantage of the IP-based approach to generating pairings is that this pricing problem requires little modification from iteration to iteration. Specifically, the only place in the model where the dual information appears is in the objective function. Therefore, the feasible region (i.e., the set of feasible pairings) does not vary. This provides two benefits. First, the effort required to set up the IP must only occur once, rather than at each iteration of the pricing problem. Second, the information learned in solving the IP under one objective function (i.e., for one set of dual values) can be used as an initial feasible solution, and thus an upper bound for subsequent instances.

3. Relaxing Variable Integrality

We conjecture that it is sufficient to restrict only the z variables to be integer

and to relax the integrality requirements of all other variables; when z is integer, all other variables will automatically be integer as well. As a result, the number of integer variables in the problem reduces from $O(|\mathcal{F}|^2 * D)$ to $O(|\mathcal{F}| * D)$.

The logic behind this conjecture is that an integer vector z , in conjunction with the balance constraints, yields either a single path or a convex combination of paths. Furthermore, each of these paths contains the same set of flights and thus the only difference between the paths is the order in which the flights appear.

We consider three cases with regards to these paths. In the first case, if there is only a single path, then this path must also satisfy all of the duty and pairing constraints and thus is a valid solution. In the second case, if there are multiple paths and all of them satisfy the duty and pairing constraints, then they are all feasible and all have the same cost (otherwise a lower-cost solution could be found by choosing only the lowest-cost path). This cannot occur, because different orderings of the flights will lead to different costs. Thus, in the third case where there are multiple paths, some of the paths must be infeasible and the feasible paths must have higher cost than the infeasible paths. We conjecture that a rearrangement of flights which leads to a feasibility violation will also lead to an increase in cost, thus ruling out the third case.

Below, we prove the conjecture for any instance where the maximum elapsed time in a duty is at most twelve hours.

Definitions:

- IP1 is the original IP formulation for generating crew pairings.
- LP1 is the LP relaxation of IP1.
- IP2 is the semi-relaxed model given by relaxing all binary variables in IP1

except for the z variables.

Lemma V.1. *If the optimal solution of LP1 is fractional then it can be decomposed into separate fractional paths. [Note that this decomposition is not necessarily unique.]*

Proof by construction. Start with a variable v_{f_i} with the optimal value of $0 < \lambda_i \leq 1$. From the balance constraints (5.25-5.32), there must exist some $x_{f_i, f_j, 1}$ or $y_{f_i, f_i, 1}$ with value $0 < \lambda_j \leq 1$. Proceeding in this fashion, ultimately some last flight $w_{f, d}$ with non-zero value must be reached. Take the minimum $\hat{\lambda}$ of all the values λ over all these variables. Clearly, this defines a path with fractional value $\hat{\lambda}$. Decrease all variables making up this path by $\hat{\lambda}$ and repeat. At each iteration, at least one variable with non-zero value will be removed from the solution and thus the procedure will terminate in a finite number of steps. \square

Lemma V.2. *At most one of the fractional paths constructed according to V.1 can be feasible (or the solution can be trivially converted to such an alternative solution).*

Proof by contradiction. Suppose LP1 leads to multiple feasible, fractional paths. Consider two such paths, X^1 and X^2 .

Case I. Suppose the paths have different costs. Without loss of generality assume that the cost of path X^1 is greater than the cost of path X^2 , i.e., $C(X^1) > C(X^2)$. The optimal solution value of LP1 can be written as

$$\lambda_1 C(X^1) + \lambda_2 C(X^2) + \dots + \lambda_n C(X^n),$$

which is strictly greater than

$$(\lambda_1 + \lambda_2) C(X^2) + \dots + \lambda_n C(X^n),$$

in contradiction of the optimality of the original solution.

Case II. Suppose the paths have the same cost, i.e., $C(X^1) = C(X^2)$. In this case, the solution

$$(\lambda_1 + \lambda_2)X^2 + \dots + \lambda_n X^n$$

is a feasible alternative with equivalent cost. \square

Lemma V.3. *A flight will be allocated to at most one duty in any optimal solution to LP1.*

Proof. This follows directly from constraint 5.24 and the integrality of z variables. \square

Lemma V.4. *Changing the order of flights within a feasible duty leads to an infeasible duty.*

Proof by contradiction. Suppose reordering the flights in duty d^1 leads to a feasible duty d^2 . d^1 is feasible so $e_{d^1} \leq \delta$. In order to change the order of flights in d^1 we need to break the link between at least one pair of consecutive flights f_1 and f_2 (with sit time s_{f_1, f_2}) and reorder the resulting flight sequences, with flight f_2 now appearing earlier (and non-sequentially) in the duty than flight f_1 and thus “shifting backwards” by twenty-four hours. If we write e_{d^1} as $B + s_{f_1, f_2}$, where B is the elapsed time of duty d^1 minus the sit time between flights f_1 and f_2 , then we have:

$$e_{d^1} = B + s_{f_1, f_2} \leq \delta$$

and so

$$\delta \geq e_{d^2} \geq B + 24 - s_{f_1, f_2}.$$

Therefore

$$(B + s_{f_1, f_2}) + (B + 24 - s_{f_1, f_2}) =$$

$$2B + 24 \leq 2\delta$$

and thus

$$B + 12 \leq \delta.$$

If

$$\delta \leq 12$$

then this is a contradiction. □

Theorem V.5. *Solving IP2 either leads to an integral solution for IP1 or leads to a fractional solution which can trivially be converted to an integer solution with equivalent cost.*

Proof by contradiction. Suppose solving IP2 yields a fractional solution. By V.1 this can be decomposed to at least two fractional paths, X^1 and X^2 . V.2 implies that only one of these can be feasible. Without loss of generality assume that X^1 is feasible and X^2 is infeasible. Note that these paths consist of the same flights according to V.3. However, the flights within some duties must be re-ordered given that the paths are distinct. By the optimality of the solution, X^2 must have lower cost than X^1 .

Case I. The dominant element of the cost function is the total cost of the duties. Therefore

$$\sum_i e_{d_i^2} < \sum_i e_{d_i^1},$$

which contradicts V.4.

Case II. The dominant element of the total cost is the TAFB. Therefore

$$\sum_i e_{d_i^2} = \sum_i e_{d_i^1}$$

and

$$T_{X^2} < T_{X^1}.$$

However, because X^2 is infeasible, it must violate the maximum elapsed time of some duty and thus $\sum_i e_{d_i^2} \neq \sum_i e_{d_i^1}$, a contradiction. \square

4. Optimality-Recognizing Cut

In the basic model, the objective function (5.6) minimizes the reduced cost of the pairing. In the column generation context, this means that the sub-problem finds the pairing with “the most negative” reduced cost. Therefore, the optimality of the master problem is established when the sub-problem returns a pairing with non-negative reduced cost. However, there are two potential drawbacks here: first, in the final iteration, the sub-problem may invest substantial time in finding the best solution (i.e., pairing with the minimum reduced cost) when all we need to know is that there is no negative reduced-cost solution. Second, due to precision issues, the sub-problem may generate a number of pairings with “practically zero” reduced cost before the master problem terminates. In order to avoid these drawbacks, we can use the following inequality in the sub-problem to manually enforce the negativity of the reduced cost:

$$(5.33) \quad C - \sum_{f \in \mathcal{F}} \sum_{d \in \mathcal{D}} \pi_f z_{f,d} \leq -\epsilon$$

In (5.33), the left hand side computes the reduced cost corresponding to the new pairing, where the right hand side is the actual tolerance for zero used in the

computer code. As a result, the optimality of the master problem is established when the sub-problem is infeasible; i.e., no negative reduced-cost pairing can be found.

Moreover, this constraint will restrict the search for the new pairing only to the “promising” portion of the feasible region. As the reduced cost is constrained to be negative, *any* feasible pairing generated in the process of finding the optimal pairing can also be used in order to improve the master problem’s objective function. Furthermore, in cases where finding the optimal pairing is not necessary, we can stop the sub-problem as soon as a feasible solution is found. As shown in Section 5.4.2, this can potentially speed up the column generation procedure.

5. Generating Multiple Columns

It is often observed that generating multiple columns at each iteration of the pricing problem can result in faster convergence. When solving an integer program with branch-and-bound, it is often the case that multiple integer-feasible solutions are encountered before the optimal solution is found. As is the case with other approaches to generating crew pairings, we can potentially improve performance by adding *all* of these columns (rather than only the optimal one) to the restricted master.

Of course, a strong branching strategy may restrict the number of intermediate solutions that are encountered in the branch-and-bound tree. This can be overcome by using a cut generation-based approach. In such an approach, after the optimal solution is found, a cut is added prohibiting this optimal pairing without excluding any other pairings from the feasible region. The problem is

then re-solved to find another viable pairing, and the process can be repeated until a sufficient number of pairings are generated. This cut could be defined as follows. Let \mathcal{V}^k be the set of v , w , x , and y variables that take value one in the optimal solution (i.e., pairing) at iteration k . Then the sum of all variables in set \mathcal{V}^k must be less than or equal to the cardinality of this set minus one. Because the variables form a unique sequence, this will eliminate the current pairing but will not rule out any other pairings.

5.4 Initial Computational Experiments

In the previous section, we presented a mixed integer programming formulation for generating crew pairings. This MIP facilitates the implementation of crew pairing for proto-typing new ideas in important research areas such as integrated planning, robust planning, and automated recovery. In order for this approach to be of value, however, it must not only be easy to implement, but also tractable in run time. The purpose of this section is to assess the tractability of the individual sub-problems, i.e., the generation of pairings. Therefore, we only solved the LP-relaxation of the master problem. Furthermore, we do not emphasize the performance characteristics of the master problem (e.g. tailing effects) because they are independent of the mechanism used to generate pairings.

5.4.1 Generating Initial Columns

In all of our computational experiments, the restricted master was initially seeded with a heuristically-generated set of columns. Specifically, we used a forward recursion to begin enumerating pairings exhaustively. However, whenever a flight had been encountered in n pairings, this flight was removed from consideration. This resulted in a diverse collection of columns. [Based on computational experiments,

we found $n = 6$ to provide a good trade-off in terms of number of columns generated versus time spent.] Finally, we also added a set of artificial columns, one per flight, to guarantee that an initial feasible solution to the restricted master could be found.

5.4.2 143-Flight Network

Generating Optimal Columns

In the first set of experiments, we considered a network of 143 flights and four crew bases, provided by a major U.S. hub-and-spoke carrier. We solved the LP relaxation of the master problem to optimality, i.e., until none of pricing problems associated with the four crew bases yielded a negative reduced-cost column.

Although a “provably optimal” solution cannot be guaranteed unless all columns with reduced cost strictly less than zero are considered, computer precision issues make this impractical. We therefore evaluated three parameters – 1, 0.1, and 0.01 – to define what constitutes a negative reduced-cost column. This parameter (ϵ) was introduced in Section 5.3.3.

The restricted master was initially seeded with a set of heuristically-generated columns and solved for the first set of dual values. We then used these dual values as input to the pricing problems – one per crew base. In subsequent iterations, if a particular crew base failed to yield a negative column, then we temporarily removed that crew base from consideration for subsequent iterations. Once all crew bases had been removed from consideration, we simultaneously reconsidered them all until no new columns could be identified (i.e., all four crew bases failed to yield a negative column for the current set of dual values).

The LP relaxation of the restricted master problem was solved using CPLEX version 10.2 with default settings. A 16 thread solver was used on a sun x4600-M2 with 8 dual core 8218 AMD Opteron cpus and 64 GB of ram with Red Hat

Precision (ϵ)	Objective Value	Total Run Time (seconds)	Number of Iterations	Number of Initial Cols	Number of Generated Cols
1	45567.2	194	160	320	401
0.1	45567.2	247	180	320	442
0.01	45567.2	252	166	320	396

Table 5.2: Generating optimal columns for 143 flight network.

Enterprise Linux version 4 as the operating system. Table 5.2 provides information about each of the three runs. Observe that the pricing problem calls on average took less than one second apiece (taking into account setup time to solve the restricted master problem at each iteration, etc.), and that the choice of precision did not substantially impact performance.

Generating Feasible Columns

The next set of experiments was motivated by the fact that it is not necessary to find the most negative reduced-cost variable in column generation – any variable with negative reduced cost is sufficient. Therefore, we ran the same data set, this time stopping when the first negative reduced-cost column was found (again, using three different parameters to define “negative”). The results are displayed in Table 5.3.

Precision (ϵ)	Objective Value	Total Run Time (seconds)	Number of Iterations	Number of Initial Cols	Number of Generated Cols
1	45567.2	186	175	320	435
0.1	45567.2	129	155	320	388
0.01	45567.2	158	160	320	388

Table 5.3: Generating feasible columns for 143 flight network.

Here it should be noted that we did not drop the objective function for the feasibility runs; instead, we set the solver parameters so that each iteration terminates as soon as a feasible solution is found. Observe that all three runs found the optimal solution (again, without substantial difference relative to the precision values).

5.4.3 329-Flight Network

The second data set that we considered (from the same U.S. carrier) consisted of 329 flights and four crew bases. Given the results of the experiments outlined in the previous section, we only conducted those experiments in which the first (rather than best) negative reduced-cost column was sought for each iteration. Table 5.4 shows the results of these experiments.

Precision (ϵ)	Objective Value	Total Run Time (seconds)	Number of Iterations	Number of Initial Cols	Number of Generated Cols
1	95102	5167	669	772	1904
0.1	95102	6694	688	772	1857
0.01	95102	7880	749	772	1543

Table 5.4: Generating feasible columns for 329 flight network.

The crew pairing problem (and set partitioning problems in general) frequently suffers from tailing effects – long periods required to prove optimality with little if any improvement in objective value – as a function of degeneracy in the master problem. That was true in our experience as well. Tables 5.5 and 5.6 show the total run times not only required to prove optimality, but also to find the optimal solution, to be within 1% of the optimal objective value, within 5%, and within 10% (in both absolute and relative terms). These results are consistent with what has been observed in the literature – high-quality solutions are found early, and then substantial time is spent on proving optimality.

Precision (ϵ)	Time to Prove Opt.	Time to Find the Opt. Sol.	Time to 1% Opt. Gap	Time to 5% Opt. Gap	Time to 10% Opt. Gap
1	5167	5054	2969	1854	1467
0.1	6694	6304	2365	1549	1252
0.01	7880	7255	2189	1322	918

Table 5.5: Details of the computation for 329 flight network (time units are seconds).

It is interesting to note that we see similar performance in the individual pricing problems themselves. Table 5.7 shows the average run time per iteration for each

Precision (ϵ)	Time to Prove Opt.	Time to Find the Opt. Sol.	Time to 1% Opt. Gap	Time to 5% Opt. Gap	Time to 10% Opt. Gap
1	100%	98%	57%	36%	28%
0.1	100%	94%	35%	23%	19%
0.01	100%	92%	28%	17%	12%

Table 5.6: Details of the computation for 329 flight network (time units are percents).

Precision (ϵ)	Ave. Time/Iter. to Prove Opt.	Ave. Time/Iter. to Find Opt.	Ave. Time/Iter. to 1%	Ave. Time/Iter. to 5%	Ave. Time/Iter. to 10%
1	7.70	7.35	6.82	7.90	8.86
0.1	9.71	9.27	5.53	6.13	6.52
0.01	10.51	9.85	4.84	5.17	4.93

Table 5.7: Average run times for 329 flight network.

of the same stages of progress. (The reason that there is an increase in time per iteration from 1% to 5% and 5% to 10% is because there are more crew bases to be investigated per iteration in the earlier iterations.)

We illustrate this further with Figure 5.1, which plots the time and optimality gap in the master problem for each iteration (using a precision value (ϵ) of 1) as well as time to solve each iteration of the sub-problem.

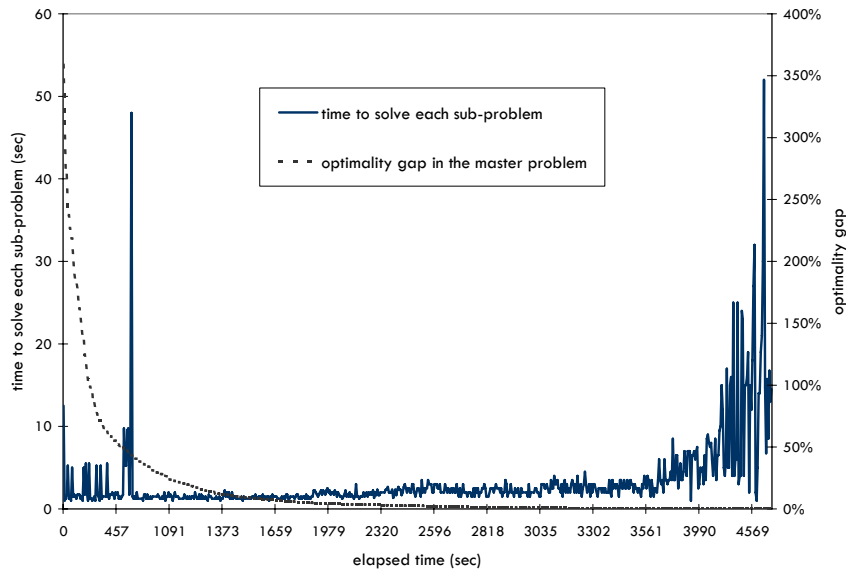


Figure 5.1: The convergence trend in the master problem for 329 flight network.

5.5 Enhanced Computational Experiments

As the size of the problem instance increases, we see significant impact on performance. This is primarily because of the growth both in number of variables and number of constraints. Table 5.8 demonstrates the differences in size between our two test instances.

Instance	no. of flights	Size of the IP		
		no. of rows	no. of columns	no. of nonzeros
1	143	424	1,406	6,104
2	329	1,312	10,633	46,417

Table 5.8: Size of the IP-based sub-problems

Note, however, that most pairings (and thus solutions to the sub-problem) contain at most ten to twenty flights, and thus the vast majority of the flights in any instance of the sub-problem will not be chosen. Furthermore, those flights with very negative dual values are unlikely to appear in promising pairings. These facts can be exploited to further improve performance. We define a *flight potential* as:

$$(5.34) \quad p_f = b_f - \pi_f$$

In other words, the potential of a flight is its flying time minus its dual value. Note that any flight f added to a pairing will increase its cost by approximately b_f and thus will impact the reduced cost by approximately p_f . Therefore, flights with more negative potentials are more likely to appear in negative reduced cost columns. We can leverage this in our approach by limiting the flight set that we consider in a single instance to only those with suitably negative potentials. More specifically, we can restrict the flights included in the IP to only a fraction of all the flights and then gradually increase the number of flights in the model if no negative-reduced cost pairing is found.

We implemented this idea in the model by updating the flight potentials at each call to the sub-problem, based on the new dual values, and adding the constraint:

$$(5.35) \quad \sum_{f:p_f > P^s} z_f = 0$$

In the above constraint, P^s is the *threshold value* for the potential of the acceptable flights in the s^{th} step of solving the pairing generation subroutine at each call to the sub-problem. In other words, constraint (5.35) excludes flight f from consideration in the s^{th} step of solving the sub-problem if the potential of that flight is greater than the threshold value, P^s . At each call to the sub-problem, the IP is initialized in the first step with only 10% of the flights. The pairing generation will continue until no negative-reduced cost pairing is found. Then in the next step, the threshold value (P^s) is adjusted so that 20% of the flights are allowed in the IP. The process of adjusting the threshold value continues until 100% of the flights are included in the IP.

This approach not only decreases the time to generate each pairing, but also reduces the solution time of the master problem significantly. Table 5.9 compares the performance of the original implementation of the pairing generator with the enhanced potential-based model.

Implementation	Objective Value	Total Run Time (seconds)	Number of Iterations	Number of Generated Cols
Original	95102	7880	749	2043
Potential-Based	95102	2065	714	1503

Table 5.9: Comparing the performance of the original and the enhanced IP for the 329-flight network.

Although this acceleration technique has dramatic impact on performance, we suggest that it can be even further improved through more sophisticated design of the algorithm with regards to issues such as the potential threshold selection and the function governing how this threshold is modified. This is a topic for future research.

Finally, we conclude by noting that this approach of limiting flights to only those with the most negative potential values does not depend on the use of IP-based approach. It could be used to improve the performance of other pairing generators as well.

5.6 Conclusions and Future Research

In this chapter, we presented a new IP-based approach to generate airline crew pairings. The main benefit of this technique is its relative ease of implementation which can be combined with the efficiency of off-the-shelf commercial solvers. This can enable researchers in the airline planning field to prototype and evaluate new ideas. The modeling ideas used in this research, such as connection and marker variables, can also be extended to other scheduling problems of similar complex structure.

Within the IP-based formulation, we have only addressed the most common rules governing feasible crew pairings. As a general rule, with regards to rules not incorporated, we have erred on the side of over-constraining. In other words, not all feasible pairings will be valid solutions to the proposed formulation. For example, U.S. airlines under FAA regulations can exceed the limit on total flying time in a duty under certain circumstances (“the 8-in-24 rule”). Specifically, if a crew flies more than eight hours in a 24 hour time window then the next rest period, called a *compensatory rest*, is required to exceed a specified lower bound that is longer than the normal limit. Thus, certain pairings that are deemed infeasible by the IP-based approach would actually be valid under these exceptions. An alternative approach, which could be used in conjunction with the existing formulation, would be to also solve the pricing problem under the less restricted conditions, then add a check to

determine whether the resulting solution was in fact a valid pairing. If not, a cut could be added to rule out this pairing and the process repeated. Evaluating this approach and identifying ways to incorporate additional levels of detail is an area for future research.

Finally, as stated, this research provides a practical component that is beneficial for the development of research in broader airline planning problems such as robust planning, integrated planning and automated recovery.

CHAPTER VI

Conclusion and Future Research

In this dissertation we proposed an analytical framework to measure and minimize the disruptive impacts of airline delays. We also introduced a new integer programming-based approach to generate crew pairings which can in turn facilitate research in the area of airline robust planning.

In Chapter III, we investigated the relationship between planned schedules and their performance when they are subject to delays. We analyzed the potential for delay propagations and defined proper metrics to quantify them. The main contribution of this section is to formalize the airline delay propagation trees and their corresponding measures. This idea, as seen in Chapter IV lays the framework for incorporating delay propagations in the airline planning process.

To make the analysis of delay propagations in Chapter III more realistic, we suggest the extension of the analysis to consider propagations due to the sharing of other common resources such as the cabin crew or connecting passengers. It is also important to develop a framework for estimating the cost of delays so that a more precise trade-off between decreased likelihood of delay propagations and increased operational costs can be performed. This, in turn, will help researchers to better incorporate robustness in the other stages of the airline planning process.

The idea of propagation trees can also be used to model disruptions that occur in other problems with an underlying networks structure. This approach can be used to define new metrics for measuring robustness of infrastructure networks such as roads, energy or water supply networks. Of course further research is needed in order to capture the subtleties of each network structure or to obtain a clear definition of “disruption” and “robustness” in each application.

In Chapter IV we introduced an optimization-based framework to minimize the potential for delays to propagate by re-allocating the slack in the flight networks. This approach can improve the operational performance of a given schedule (and subsequently, passenger convenience and productivity) by considering the effects of each delay on down-stream flights.

This model can be improved by incorporating a more realistic objective function which takes into account factors such as the planned costs and anticipated operational costs. Other factors such as the size of the aircraft (and subsequently, number of passengers on each flight) or number of passengers with lost itineraries can also be considered in the objective function to represent a more realistic interpretation of delays. However, as mentioned before, quantifying the indirect (and often invisible) costs associated with disrupted passengers is not trivial.

Another possible extension of this model is to consider recovery decisions (such as canceling flights, swapping aircraft, and calling in reserve crews) and their impacts on the operational performance of a planned schedule under disruptions. These actions can be considered as the “second-stage” decisions in addition to the “first-stage” decisions (slack re-allocation). This approach will result in a two-stage stochastic (or robust) planning problem that can integrate the scheduling and recovery decisions. Further research in this area is another possible extension to the models introduced

in Chapter IV.

In Chapter V, we presented a new IP-based approach to generate airline crew pairings. This model can facilitate research in the area of robust airline planning by helping researchers to implement the crew pairing module.

The model presented in Chapter V considers only the most basic rules governing feasible crew pairings. Incorporating more sophisticated regulations (such as “8-in-24” rule) can be a possible future extension of this model.

It is also important to explore the possibility of using “connection” and “marker” variables defined in Chapter V in other scheduling applications.

BIBLIOGRAPHY

BIBLIOGRAPHY

- [1] J. Abara. Applying integer linear programming to the fleet assignment problem. *Interfaces*, 19:20–38, 1989.
- [2] A. Abdelghany, G. Ekollu, R. Narasimhan, and K. Abdelghany. A proactive crew recovery decision support tool for commercial airlines during irregular operations. *Annals of Operations Research*, 127:309–331, 2004.
- [3] J.P. Arabeyre, J. Fearnley, F.C. Steiger, and W. Teather. The airline crew scheduling problem: A survey. *Transportation Science*, 3(2):140–163, 1969.
- [4] E. Balas and M. Padberg. Set partitioning: A survey. *SIAM Review*, 18(4):710 – 760, 1976.
- [5] C. Barnhart, A. Cohn, E. Johnson, D. Klabjan, G. Nemhauser, and P. Vance. *Handbook of Transportation Science*, chapter Airline Crew Scheduling. Kluwer’s International Series, 2003.
- [6] C. Barnhart, L. Hatay, and E. Johnson. Deadhead selection for long-haul crew pairing problem. *Operations Research*, 43(3):491 – 499, 1995.
- [7] C. Barnhart and K. Talluri. *Design and Operation of Civil and Environmental Engineering Systems*, chapter Airline Operations Research, pages 435–469. John Wiley and Sons, New York, 1997.
- [8] R. Beatty, R. Hsu, L. Berry, and J. Rome. Preliminary evaluation of flight delay propagation through an airline schedule. 1998. Proceedings of the 2nd USA/Europe Air Traffic Management R&D Seminar.
- [9] M. Berge. Timetable optimization: Formulation, solution approaches, and computational issues. pages 341–357, 1994. AGIFORS proceedings, Zandvcort, The Netherlands.
- [10] S. Bratu and C. Barnhart. Flight operations recovery - new approaches considering passenger recovery. *Journal of Scheduling*, 9:279–298, 2006.
- [11] E.K. Burke, P. De Causmaecker, G. De Maere, J. Mulder, M. Paelinck, and G. Vanden Berghe. A multi-objective approach for robust airline scheduling. *Computers and Operations Research*, (to appear), 2007.
- [12] M. Clarke, L. Lettovský, and B. Smith. *Handbook of Airline Operations*, chapter The Development of the Airline Operations Control Center, pages 131–147. Aviation Week, 2000.
- [13] A. M. Cohn and C. Barnhart. Improving crew scheduling by incorporating key maintenance routing decisions. *Operations Research*, 51:387–396, 2003.
- [14] J. Cordeau, G. Stojković, F. Soumis, and J. Desrosiers. Benders decomposition for simultaneous aircraft routing and crew scheduling. *Transportation Science*, 35(4).
- [15] M. Desrochers and F. Soumis. A generalized permanent labeling algorithm for shortest path problems with time windows. *INFOR*, 26:191 – 212, 1988.

- [16] M. Ehrgott and D. M. Ryan. Constructing robust crew schedules with bicriteria optimization. *Journal of Multi-Criteria Decision Analysis*, 11(3):139150, 2002.
- [17] T. Fahle, U. Junker, S. Karisch, N. Kohl, M. Sellmann, and B. Vaaben. Constraint programming based column generation for crew assignment. *Journal of Heuristics*, 8:59 – 81, 2002.
- [18] B. Fuhr. Robust flight scheduling - an analytic approach to performance evaluation and optimization. Technical report, 2007. Clausthal University of Technology - Lufthansa Airlines.
- [19] R.S. Garfinkel and G.L. Nemhauser. Optimal political districting by implicit enumeration techniques. *Management Science*, 16(8):495 – 508, 1970.
- [20] P.C. Gilmore and R.E. Gomory. A linear programming approach to the cutting stock problem. *Operations Research*, 11(6):863 – 888, 1961.
- [21] K.L. Hoffman and M. Padberg. Solving airline crew scheduling problems by branch and cut. *Management Science*, 39(6):657 – 682, 1993.
- [22] C. Hsiao and M. Hansen. An econometric analysis of us airline flight delays with time-of-day effects. 2006. Proceedings of TRB 2006 Annual Meeting.
- [23] <http://www.airlines.org>. Air transport association.
- [24] <http://www.transtats.bts.gov>. Bureau of transportation statistics.
- [25] M. Janic. Modeling the large scale disruption of an airline network. *Journal of Transportation Engineering*, 131(4):249 – 260, 2005.
- [26] E. L. Johnson, G. L. Nemhauser, and M.W.P. Savelsbergh. Progress in linear programming-based algorithms for integer programming: An exposition. *INFORMS Journal on Computing*, 12(1):2–23, 2000.
- [27] D. Klabjan, E. Johnson, G. Nemhauser, E. Gelman, and S. Ramaswamy. Airline crew scheduling with time windows and plane-count constraints. *Transportation Science*, 36(3):337–348, 2002.
- [28] D. Klabjan, E. L. Johnson, and G. L. Nemhauser. Airline crew scheduling with regularity. *Transportation Science*, 35(4):359–374, 2001.
- [29] D. Klabjan, E. L. Johnson, and G. L. Nemhauser. Solving large airline crew scheduling problems: Random pairing generation and strong branching. *Computational Optimization and Applications*, 20:73 – 91, 2001.
- [30] D. Klabjan and K. Schwan. Airline crew pairing generation in parallel. Technical report, Georgia Institute of Technology, 2001. TLI/LEC-99-02.
- [31] N. Kohl and S.E. Karisch. Airline crew rostering: Problem types, modeling, and optimization. *Annals of Operations Research*, 127:223–257, 2004.
- [32] N. Kohl, A. Larsen, J. Larsen, A. Ross, and S. Tiourine. Airline disruption management perspectives, experiences and outlook. *Journal of Air Transport Management*, 13:149–162, 2007.
- [33] S. Lan, J.P. Clarke, and C. Barnhart. Planning for robust airline operations: Optimizing aircraft routings and flight departure times to minimize passenger disruptions. *Transportation Science*, 40(1), 2006.
- [34] M. Lapp, S. AhmadBeygi, A.M. Cohn, and O. Tsimhoni. A recursion-based approach to simulating airline schedule robustness. Technical report, 2008. Submitted to proceedings of the 2008 Winter Simulation Conference.

- [35] S. Lavoie, M. Minoux, and E. Odier. A new approach for crew pairing problems by column generation with an application to air transportation. *European Journal of Operational Research*, 35:45 – 58, 1988.
- [36] L. Lettovský, E.L. Johnson, and G.L. Nemhauser. Airline crew recovery. *Transportation Science*, 34(4):337–348, 2000.
- [37] A. Levin. Scheduling and fleet routing models for transportation systems. *Transportation Science*, 5:232–255, 1971.
- [38] M. Lohatepanont and C. Barnhart. Airline schedule planning: Integrated models and algorithms for schedule design and fleet assignment. *Transportation Science*, 38(1):19–32, 2004.
- [39] A. Makri and D. Klabjan. A new pricing scheme for airline crew scheduling. *INFORMS Journal on Computing*, 16(1):56 – 67, 2004.
- [40] R. E. Marsten. An algorithm for large set partitioning problems. *Management Science*, 20(5):774 – 787, 1974.
- [41] A. Mercier and F. Soumis. An integrated aircraft routing, crew scheduling and flight retiming model. *Computers and Operations Research*, 34(8):2251–2265, 2007.
- [42] George L. Nemhauser and Laurence A. Wolsey. *Integer and combinatorial optimization*. Wiley-Interscience, New York, NY, USA, 1988.
- [43] B. Rexing, C. Barnhart, T. Kniker, A. Jarrah, and N. Krishnamurthy. Airline fleet assignment with time windows. *Transportation Science*, 34(1):120, 2000.
- [44] J.M. Rosenberger, E. Johnson, and G. Nemhauser. A robust fleet-assignment model with hub isolation and short cycles. *Transportation Science*, 38(3):357–369, 2004.
- [45] J.M. Rosenberger, A.J. Schaefer, D. Goldsman, A.J. Kleywegt, E.L. Johnson, and G.L. Nemhauser. A stochastic model of airline operations. *Transportation Science*, 36(4):357–377, 2002.
- [46] N. Rupp. Investigating the causes of flight delays. 2007. Working paper, Department of Economics, East Carolina University.
- [47] D.M. Ryan and J.C. Falkner. On the integer properties of scheduling set partitioning models. *European Journal of Operational Research*, 35:442 – 456, 1988.
- [48] R. Sandhu and D. Klabjan. Integrated airline fleet and crew-pairing decisions. *OPERATIONS RESEARCH*, 55(3):439–456, 2007.
- [49] A. Schaefer, E. Johnson, J. Kleywegt, and G. Nemhauser. Airline crew scheduling under uncertainty. *Transportation Science*, 39(3):340–348, 2005.
- [50] S. Shebalov and D. Klabjan. Robust airline crew pairing: Move-up crews. *Transportation Science*, 40(3):300–312, 2006.
- [51] B. Smith and E. Johnson. Robust airline fleet assignment: Imposing station purity using station decomposition. *Transportation Science*, 40(4):497–516, 2006.
- [52] G. Stojković, F. Soumis, J. Desrosiers, and M. M. Solomon. An optimization model for a real-time flight scheduling problem. *Transportation Research Part A*, 36:779–788, 2002.
- [53] M. Stojković and F. Soumis. An optimization model for the simultaneous operational flight and pilot scheduling problem. *Management Science*, 47(9):1290–1305, 2001.
- [54] D. Teodorovic and E. Krmar-Nozic. Multicriteria model to determine flight frequencies on an airline network under competitive conditions. *Transportation Science*, 23(4):14–25, 1989.

- [55] Y. Tu, M. Ball, and W. Jank. Estimating flight departure delay distributions - a statistical approach with long-term trend and short-term pattern. 2006. Working paper, Department of Decision and Information Technologies, University of Maryland.
- [56] P. Wang, L. Schaefer, and L. Wojcik. Flight connections and their impacts on delay propagation. Technical report, 2003. MITRE.
- [57] S. Yan and Y. Tu. A network model for airline cabin crew scheduling. *European Journal of Operational Research*, 140:531 – 540, 2001.
- [58] S. Yan and D.H. Yang. A decision support framework for handling schedule perturbation. *Transportation Research Part B*, 30(6):405–419, 1996.
- [59] J. Yen and J. Birge. A stochastic programming approach to the airline crew scheduling problem. *Transportation Science*, 40(1):3–14, 2006.
- [60] G. Yu and X. Qi. *Disruption management: Framework, Models and Applications*. World Scientific Publishing, 2004.