

EMPIRICAL INVESTIGATIONS OF PROPERTIES
OF ROBUST AIRCRAFT ROUTING MODELS

BY

GUANQUN LI

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Industrial Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2015

Urbana, Illinois

Advisor:

Assistant Professor Lavanya Marla

ABSTRACT

The airline schedule planning process is an important component of airline operations, and it involves considerably complex problems. This research focuses on the aircraft routing phase. We introduce the concept of robustness in aircraft routing problems, and find solutions that can stand uncertainty.

We categorize the delays in flight operations into two components – independent delay and propagated delay. In the data driven approach, independent delay can be regarded as constant, but propagated delay can be worked on. An example of aircraft swap is given to show that aircraft routing can potentially reduce the flight delays. To solve robust aircraft routing problems, we propose a list of formulations. They are in three categories – Lan, Clarke, Barnhart’s approach, chance-constrained programming approach, and extreme value approach.

We conduct experiments with two airline networks – a 50-flight network and a 165-flight network. The K -fold cross validation approach is incorporated into aircraft routing problems to eliminate overfitting. According to the three evaluation metrics – on time performance, average total propagated delay and passenger disruptions, several good formulations are identified, which are recommended for airline schedule planners. We also explain the reasons behind the solution differences.

TABLE OF CONTENTS

Chapter 1: Introduction	1
1.1 Background of Airline Schedule Planning	1
1.2 Delays in Airline Operations	3
1.3 Motivation	6
1.4 Outline of Thesis	8
Chapter 2: Literature Review	10
2.1 Robustness of Airline Schedule Planning	10
2.1.1 Airline Schedule Planning Literature Review	10
2.1.2 Robust Aircraft Routing Literature Review	13
2.1.2 Robustness Literature Review	15
2.2 Aircraft Routing Problem Fundamentals	17
2.3 Evaluation Metrics	21
Chapter 3: Robust Aircraft Routing Models	24
3.1 The Deterministic Aircraft Routing Model	24
3.2 Lan, Clarke, Barnhart's Approach	28
3.3 Chance-Constrained Programming Approach	29
3.4 Extended Chance-Constrained Programming Approach	32
3.5 Bertsimas and Sim's Extreme Value Approach	33
3.6 Delta Extreme Value Approach	35
3.7 Delta Objective Extreme Value Approach	36
Chapter 4: Evaluation and Result Analysis	40
4.1 Experimental Set-up	40
4.1.1 Description of the Raw Data Sources	40
4.1.2 Pre-processing of the Data	41
4.1.3 Description of Selected Networks	42
4.1.4 Generating Strings in each Network	43
4.1.5 Generating Inputs to Mathematical Models	43
4.1.6 Running Robust Aircraft Routing Models	44

4.1.7 K-fold Cross Validation	45
4.2 Analysis of Results.....	46
4.2.1 Computation Time	46
4.2.2 Comparison of Solution Quality.....	47
4.2.3 Comparison of Model Performances	50
Chapter 5: Conclusions	53
Appendix A: Independent Delay Data.....	55
A.1 Independent Delay Simulation for the 50-flight network	55
References	57

CHAPTER 1 INTRODUCTION

1.1 Background of Airline Schedule Planning

The airline schedule planning process is an important component of airline operations, and it involved high complexity. The airline schedule comprises a number of elements. A normal-sized airline schedule is a very large-scale network, with hundreds of flights per day. Modeled mathematically, the network contains hundreds of nodes (representing airports at various points in time) and millions of arcs (representing flights between these airports). In addition, we need to take into account many factors, such as airport gates, slots, aircraft types, crew restrictions, aircraft maintenance requirements and passenger demands. Therefore, airline schedule planners decompose the problem into four subproblems: (i) Schedule Design, (ii) Fleet Assignment, (iii) Aircraft Maintenance Routing, and (iv) Crew Scheduling. Next we briefly introduce each problem [1].

(i) Schedule Design

The objective of the Schedule Design problem is to determine a set of flight legs, with specified origin, destination, scheduled departure time, and scheduled arrival time. The main design criteria is the market demand estimation. It usually requires the collaboration of many business units of an airline to design the schedule. Therefore, though the main goal is to optimize the estimated profit of this schedule, this problem is rarely solved using mathematical models.

(ii) Fleet Assignment

Given an airline schedule from the previous subproblem, we need to determine the type of aircraft that will operate each flight leg, taking into account the total number of aircraft of each fleet type available. Airline schedule planners consider both economic profitability and operational feasibility. They minimize the total cost, which includes the cost of operating a flight leg with a specified type of aircraft and spill cost (the opportunity cost of having insufficient seating capacity to satisfy passenger demands). Constraining the assignment of aircraft type to flights is the total number of aircraft of each type available, and the requirement that each aircraft have a feasible itinerary. This is verified by creating a balanced network, where the inflow and outflow of each node is balanced for each type of aircraft.

(iii) Aircraft Maintenance Routing

In practice, each aircraft has to enter maintenance after a limited number of flying hours. Given a flight schedule and a fleet assignment, the Aircraft Maintenance Routing subproblem ensures that each individual aircraft, described by its tail number, has a feasible route between two maintenance periods. Constraints in this mathematical optimization model are that each flight leg should be operated by one and exactly one aircraft, aircraft flow balance should hold, and the number of aircraft used is less than the number available.

(iv) Crew Scheduling

Given the solutions to the three previous problems, Crew Scheduling is the final subproblem. It aims at assigning cockpit crew and cabin crew to all the flight legs at the least possible cost. This problem is the most complex among the four subproblems due to various labor restrictions and mutual agreements between airline companies and employees. Because of complexity, crew scheduling is divided into two subproblems, crew pairing and crew assignment. In the crew pairing problem, we create multi-day sequences of flight legs with lower costs, which are called pairings. These pairings must satisfy labor restrictions. In the crew assignment problem, we combine the pairings into month long crew schedules, which are called bidlines or rosters, then the schedules are assigned to each crew member according to each one's preferences.

In the past, the airline scheduling problems have been mostly solved in sequence, as described above. However, solving four problems sequentially typically gives a suboptimal solution. Therefore, in the past few decades, airline schedule planners have made lots of efforts to integrate some problems. On the other hand, due to high level of complexity, these problems are mostly solved assuming that the flights will be operated as planned. Ignoring the potential disturbances will cause the flight schedule to be vulnerable to delays and cancellations. Therefore it calls for a *robust airline schedule planning* process.

1.2 Delays in Airline Operations

In practice, airline schedule planners used to solve the four subproblems of airline scheduling as deterministic process. That is, they optimize the schedule based upon the assumption that the aircraft depart and arrive on the exact time as planned. In

reality, it is almost never the case. Therefore, factoring in the inevitability of delays and disruptions is a factor of high importance.

According to the on-time performance statistics given by Department of Transportation (DoT), the percentage of aircraft arrival delays has been around 20% from the year 2005 to the year 2014 (see Table 1).

Year	Ontime Arrivals	Ontime (%)	Arrival Delays	Delayed (%)	Flights Cancelled	Cancelled (%)	Diverted	Flight Operations
2005	5,526,773	77.40%	1,466,065	20.53%	133,730	1.87%	14,027	7,140,595
2006	5,388,265	75.45%	1,615,537	22.62%	121,934	1.71%	16,186	7,141,922
2007	5,473,439	73.42%	1,804,028	24.20%	160,809	2.16%	17,182	7,455,458
2008	5,330,294	76.04%	1,524,735	21.75%	137,432	1.96%	17,265	7,009,726
2009	5,127,157	79.49%	1,218,288	18.89%	89,377	1.39%	15,463	6,450,285
2010	5,146,504	79.79%	1,174,884	18.21%	113,255	1.76%	15,474	6,450,117
2011	4,845,032	79.62%	1,109,872	18.24%	115,978	1.91%	14,399	6,085,281
2012	4,990,223	81.85%	1,015,158	16.65%	78,862	1.29%	12,519	6,096,762
2013	4,990,033	78.34%	1,269,277	19.93%	96,012	1.51%	14,160	6,369,482
2014	4,437,850	76.25%	1,240,528	21.32%	126,984	2.18%	14,449	5,819,811

Table 1 On-time Performance from 2005 to 2014

Delays have a negative economic impact. According to the Ball et al. [2], delays result in costs for airlines, for passengers, and in terms of lost demands. It estimated that the annual cost of U.S. flight delays is \$31 billion. Airlines for America [3] also estimated that in 2013, the cost of aircraft block (taxi plus airborne) time for U.S. passenger airlines was \$76.22 per minute.

Because of such high economic loss caused by airline delays, the planners should take into consideration flight schedule feasibility and profitability, as well as robustness. According to Federal Aviation Administration (FAA), we can categorize the detailed reasons of flight delays into five main types [4].

- Air Carrier Delay

Delay is within the control of the air carrier. Examples of occurrences that may determine carrier delay are: aircraft cleaning, awaiting the arrival of connecting passengers or crew, baggage, cargo loading, crew legality (pilot or attendant rest), fueling, handling disabled passengers, late crew, oversales, slow boarding or seating delays.

- Late Arrival Delay

Arrival delay at an airport due to the late arrival of the same aircraft from a previous airport. The ripple effect of an earlier delay at downstream airports is referred to as delay propagation.

- NAS Delay

Delay that is within the control of the National Airspace System (NAS) may include: non-extreme weather conditions, airport operations, heavy traffic volume, air traffic control, etc.

- Security Delay

Security delay is caused by evacuation of a terminal or concourse, re-boarding of aircraft because of security breach, inoperative screening equipment and/or long lines in excess of 29 minutes at screening areas.

- Weather Delay

Weather delay is caused by extreme or hazardous weather conditions that are forecasted or manifest themselves on point of departure, enroute, or on point of arrival.

From January to December 2014, 23.75% of flights are delayed, and 34.6% of the flight delays are caused by Late Arrival Delay (Chart 1). More importantly, among

the delay reasons, airline planners can only control the Air Carrier Delays and Late Arrival Delays. Therefore, this thesis mainly focuses on the mathematical models to minimize Late Arrival Delays.

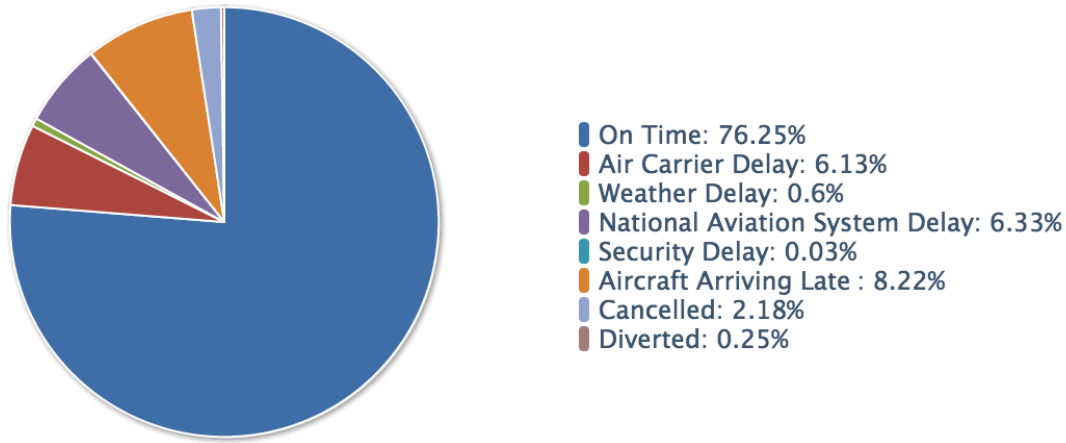


Chart 1 Airline Delay Cause Statistics in December 2014

Source: Bureau of Transportation Statistics, DoT

1.3 Motivation

Conventionally, optimization problems are solved assuming that the input data is deterministic. Models are typically solved using *mean values*, *best-guess values* or *worst-case values*. But in many occasions, these formulations fail to generate satisfactory solutions. These kinds of optimization models are called *nominal* models [5]. *Robust optimization*, an approach that specifically considers model vulnerability, is designed to address the problems of nominal models. The solutions produced from such models are called *robust solutions*. This thesis mainly focuses on

different types of robust models, and focuses on demonstrating their effectiveness through the Aircraft Maintenance Routing problem.

In the previous section, we have seen that more than one third of the flight delays are caused by Late Arrival Delays. This indicates that the delay from a previously late arriving aircraft is propagated to the following flight operated by that aircraft. The aircraft routing problem determines the sequence of flights to be operated by the same aircraft, and because of that, the aircraft routing solution directly impacts the late arrival delays. Also, conventionally, the purpose of the Aircraft Routing problem is to find a feasible solution that is amenable to maintenance rather than to find an optimal solution with respect to a specific objective function. Therefore, it is easy for aircraft routing planners to model the consequence of late arrivals, as well as to perform various other kinds of experiments and then analyze different solutions. Due to this feature, this thesis focuses on the aircraft maintenance routing problem. This thesis proposes several models, with each modeling different aspects of flight delays and disruptions. The robust models consider the probability distribution of flight delay performance, and include aspects of the distribution into the formulation (such as quantiles and worst-case values), and thus they can be better in terms of dealing with potential delays.

In earlier practice, the responses to flight delays were reactive, which means, after a delay or disruption occurs, recovery actions will be implemented to mitigate the effects of the disruption and bring the schedule back to the plan. This can be usually far more expensive and complex than a pro-active approach. Robust methods are pro-active, that is, seek to build solutions that are more robust a priori (though may be more expensive), but will reduce the sum of the planning and recovery costs.

To analyze the different solutions derived from the different formulations, this thesis utilizes a set of evaluation metrics to understand the advantage and disadvantage of each aircraft routing solution.

1.4 Outline of Thesis

In Chapter 2, we present a review of the literature on the topic of robustness in airline scheduling. First, we briefly browse the important results in the field of robust airline scheduling problems. Second, we have a closer look at the aircraft maintenance routing problem. We explain the modeling concept and formulations, followed by a discussion of existing work in the area. Third, we summarize the general evaluation criteria used by researchers, and then we outline the evaluation metrics used by this thesis to analyze the solutions.

In Chapter 3, we present three categories of robust aircraft routing models. We begin with the deterministic approach, then we move on to Lan, Clarke, Barnhart's robust approach; then Charnes and Cooper's Chance-Constrained Programming approach; and finally Bertsimas and Sim's extreme-value based robust optimization method. In each category we present multiple models that capture different aspects of the problem. To evaluate the solutions that arise from these models, we use a 5-fold cross validation approach that avoids overfitting and allows for generalizability of our results.

In Chapter 4, we discuss the experimental setup for two real-world instances of different sizes. We then explain the solution process and analyze the solutions from the various models in terms of the evaluation metrics.

In Chapter 5, we summarize our findings.

CHAPTER 2 LITERATURE REVIEW

2.1 Robustness of Airline Schedule Planning

2.1.1 Airline Schedule Planning Literature Review

Most of the literature on robust airline scheduling focuses on identifying attributes of the subproblem or subproblems of interest that contribute to robustness of the schedule. Most approaches then define optimization-based or simulation-based approaches that maximize the presence of these attributes and increase robustness. Such attributes are defined in different ways in terms of move-up crews, hub connectivity, propagated delays, station purity, etc., as we describe below.

Shebalov and Klabjan [6] build a robust model on crew schedule planning. They introduce two objectives - minimizing the crew cost, and maximizing the number of move-up crews – which means the crews that can potentially be swapped in operations. They use delayed column generation, and Lagrangian decomposition for solving the restricted master problem. Their experimental results show that a robust crew scheduling solution sacrifices the total crew cost.

Rosenberger, Johnson, and Nemhauser [7] extend the fleet assignment model with the concept of cancellation cycles and hub connectivity. A cancellation cycle is a sequence of flights that begins and ends at the same airport. Hub connectivity is the number of legs in a rotation that are in a route that begins at a hub, ends at a different hub, and only stops at spokes in between. They point out that a fleet assignment and aircraft rotation with many short cancellation cycles is more robust to a flight cancellation. Low hub connectivity also mitigates the impact of

propagated disruptions from one hub to others. They use a simulation of airline operations, SimAir, to solve the assignments.

Schaefer et al. [8] consider algorithms for finding crew schedules that perform well in practice. They introduce two ways of measuring pilot compensation - planned cost of a crew schedule and operational cost of a crew schedule. Planned cost is a deterministic value traditionally used. They calculate operational cost, a random variable which features planned cost, and finally minimize the expected operational cost. They provide a lower bound on the cost of an optimal crew schedule in operations, and prove that their method gives the expected cost very close to the lower bound.

Yen and Birge [10] consider the crew scheduling problem with uncertainty. They first formulate it as a stochastic integer programming model, and then transfer it to a nonlinear recourse model. To solve the problem, flight-pair branching algorithm is used. It branches simultaneously on multiple variables by allowing or disallowing key flight pairs where crews switch planes. They provide hierarchy for flight pairs to branch, based on the delay costs. Their method results in overall savings in the expected cost of a crew schedule when disruptions are considered.

Smith and Johnson [11] extend fleet assignment models by imposing station purity, limiting the number of fleet types allowed to serve each airport in the schedule. For the computational efficiency, they use station decomposition – a column generation approach, to solve the fleet assignment problem. They further improve the performance of station decomposition by developing a primal-dual method. Additionally, they develop a “fix-and-price” heuristic to efficiently find integer

solutions, because station decomposition solutions can be highly fractional. Their estimation shows there can be significant reduction in cost for a major U.S. domestic airline by applying station purity.

Burke et al. [12] propose a mimetic approach for multi-objective improvement of robustness objectives in airline schedules. They consider two objectives – schedule reliability and schedule flexibility. Their variables characterize flight retiming and aircraft rerouting simultaneously, subject to a fixed fleet assignment. They approximate the Pareto optimal front by applying a multi-meme mimetic algorithm. The experiment is based on real world schedules from KLM Royal Dutch Airlines. They are able to obtain schedules with significant improvements for the considered objectives. Rigorous sensitivity analysis of the results shows that the influence of the schedule reliability is dominant and that increased schedule flexibility could improve the operational performance.

The multi-objective approach is extended in Burke et al. [13]. This approach maintains a good balance between the individual robustness objectives that maximize the operational performance of the schedule. They adopt time window approaches for incremental and integrated multi- objective improvement of robustness objectives in airline schedules. Their simulation result shows the reliability of the scheduled times has a dominant influence on the punctuality of the schedule. The flexibility of the schedule was shown to become more important for smaller schedules. Balance between the reliability at hub and spoke stations results in an improved operational performance of the overall schedule.

Sohoni, Lee, and Klabjan [15] provide two service level metrics – flight service level and network service level. Flight service level is similar to the on-time performance measure of the U.S. DoT, and network service level features completion of passenger itineraries. Then they develop a stochastic integer programming formulation that maximizes expected profit while ensuring the two service levels. They apply cut generation algorithm to solve the models.

Arikan, Deshpande, and Sohoni [20] develop stochastic models, use empirical data, to analyze the propagation of delays through air-transportation networks. Based on the analysis, they make policy recommendations regarding managing bottleneck resources in the air-travel infrastructure. They concluded that the DOT on-time metric can significantly inflate true on-time performance and can be misleading, particularly to passengers with short connections. If providing accurate information to passengers is desirable, then the DOT OTP metric should be modified so that on-time really means “on-time”. A careful cost benefit analysis of this proposal needs to be conducted.

2.1.2 Robust Aircraft Routing Literature Review

Lan, Clarke, and Barnhart [9] present two new approaches to minimize passenger disruptions and achieve robust airline schedule plans. The first approach involves aircraft routing. They formulate a mixed-integer programming problem with stochastically generated inputs to reduce delay propagation. The second involves retiming flight departure times. It considers passengers who miss their flight legs due to insufficient connection time. Their objective is to minimize the number of

passenger misconnections, realized by retiming the departure times of flight legs within a small time window.

Weide, Ryan, and Ehrgott [14] develop an iterative approach to robust and integrated aircraft routing and crew scheduling. To bypass the computational difficulty, they start from a minimal cost solution, and then produce a series of solutions which are increasingly robust. The program stops when the crew penalty exceeds the predetermined threshold. Their algorithm can now generate solutions for the aircraft routing and two crew pairing problems in one integrated procedure.

Marla and Barnhart [16] compare the results to aircraft maintenance routing problems by using the chance-constrained programming approach of Charnes and Cooper [17], the extreme value approach of Bertsimas and Sim [18], [19], with the results of Lan et al. [9]. They perform an empirical experiment and propose a set of metrics to evaluate the results. They also extend the formulation to general network-based resource allocation problems.

Yan and Kung [21] extend the robust aircraft routing problem by Lan et al. [9]. The objective is to minimize the maximum possible total propagated delay, assuming flight leg delays lie in a pre-specified uncertainty set. They propose an exact decomposition solution approach under a column-and-row generation framework. By using delay correlation, their robust model outperforms the state-of-the-research stochastic optimization approach in reducing standard deviation and maximum value of total propagated delay.

Robust airline scheduling approaches proposed in the literature (Section 2.1.1 and 2.1.2) apply to different phases of the airline schedule planning process. They are summarized in Table 2.

	Schedule Design	Fleet Assignment	Aircraft Routing	Crew Scheduling
Shebalov and Klabjan (2006) [6]				•
Rosenberger et al. (2004) [7]		•	•	
Schaefer et al. (2005) [8]				•
Yen and Birge (2006) [10]				•
Smith and Johnson (2006) [11]		•		
Burke et al. (2009) [12]	•		•	
Burke et al. (2015) [13]	•		•	
Sohoni, Lee, and Klabjan (2010) [15]	•			
Arikan et al. (2012) [20]	•			
Lan, Clarke, and Barnhart (2006) [9]	•		•	
Weide, Ryan, and Ehrgott (2009) [14]			•	•
Marla and Barnhart (2010) [16]			•	
Yan and Kung (2014) [21]			•	

Table 2 Robust Schedules Literature by Planning Phase

2.1.3 Robustness Literature Review

There are also some works that provide important mathematical background for robust airline scheduling problems, but not directly solving them. They are listed as follows.

Bertsimas and Sim [18] study robust models for some discrete optimization problems. They present a model for cost uncertainty in which each coefficient is allowed to vary within an interval, with no more than a limited number of coefficients allowed to vary. In this way, the robust version of a combinatorial

problem may be solved by solving no more than $n + 1$ instances of the underlying, nominal problem. This result extends to approximation algorithms for combinatorial problems. For network flow problems, the above model can be applied and the robust solution can be computed by solving a logarithmic number of nominal, network flow problems.

Bertsimas and Sim [19] use cardinality constrained uncertainty to address robust linear optimization. They define a family of polyhedral uncertainty sets that encode a budget of uncertainty in terms of cardinality constraints: the number of parameters of the problem that are allowed to vary from their nominal values. By relaxing and taking the dual of the inner maximization problem, one can transfer the cardinality problem to a linear formulation, and therefore the problem is tractable, and moreover can be cast equivalently as a linear optimization problem.

Bertsimas and Thiele [22] assume lack of perfect information about system parameters. Accordingly, they develop two methods to solve decision-making models under uncertainty – robust optimization and data driven optimization. In robust optimization, random variables are modeled as uncertain parameters belonging to a convex uncertainty set and the decision-maker protects the system against the worst case within that set. Data-driven optimization uses observations of the random variables as direct inputs to the mathematical programming problems. They take advantage of some examples in inventory management and portfolio management to describe the robust optimization paradigm in detail, and to address the issue of constructing uncertainty sets using historical realizations of the random variables.

Limited work has been done on applying these models in the airline scheduling context. These include Marla and Barnhart [16] and Yan and Kung [21]. Marla and Barnhart model the robustness according to chance-constrained approach and extreme-value approach; Yan and Kung extend Lan et al.'s approach by adding modeling uncertainty set. This thesis uses 5-cross validation approach, which incorporates Lan et al.'s approach, chance-constrained approach and extreme-value approach, but further improves the solutions of these approaches by avoiding overfitting.

2.2 Aircraft Routing Problem Fundamentals

We focus on the aircraft routing step of the airline scheduling process. The aircraft maintenance routing problem is to design a set of sequential flight legs, which can also be called routes or routings, to be operated by each aircraft, such that each aircraft is subject to regular and periodic maintenance checks.

- Cover constraints

Each flight leg in the schedule is operated by exactly one aircraft.

- Balance constraints

The number of aircraft entering an airport is the same as the number departing from the airport

- Count constraints

The number of aircraft used is limited by the number of available aircrafts

Before we demonstrate how aircraft routing can change the on-time performance (OTP), we first introduce the concept of propagated delays. Assume there is a pair of

connecting flights. Flight f_1 departs from airport 1 and arrives at airport 2, then flight f_2 departs from airport 2 and arrives at airport 3 (Figure 1). f_1 and f_2 are operated by the same aircraft. Minimum turn time is required for this aircraft. And slack time is the difference of aircraft connection time and minimum turn time. If no delay occurs, f_1 arrives at time A_1 , and f_2 arrives at time B_1 , and some slack time can be used for connection. However, if flight f_1 is delayed, denoted as f_1' , and the amount of delay exceeds the slack time, then the departure of f_2 is consequentially delayed, denoted as f_2' .

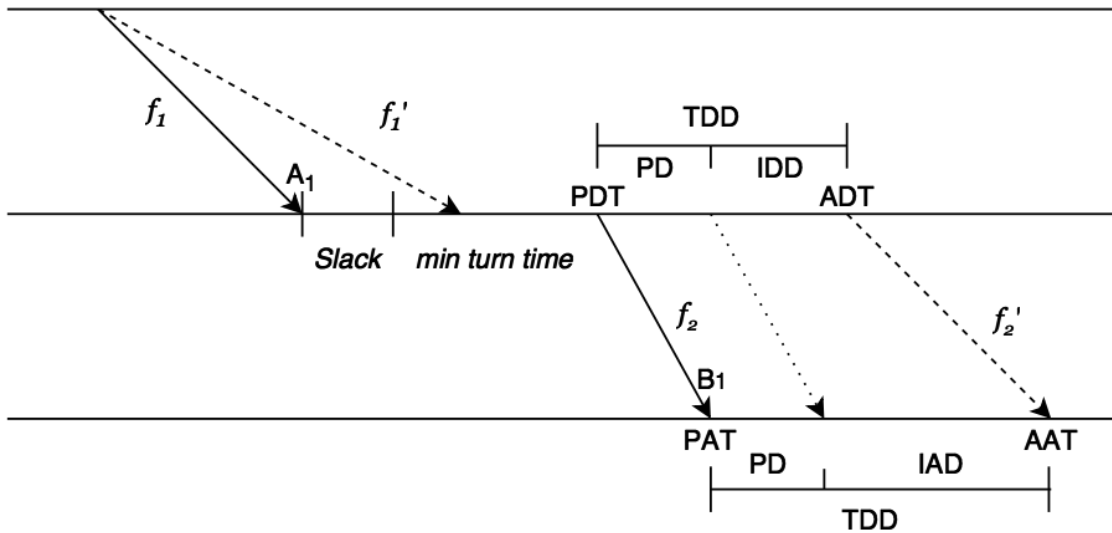


Figure 1 Demonstration of Propagated Delay

Glossary:

PDT: Planned departure time

ADT: Actual departure time

PAT: Planned arrival time

AAT: Actual arrival time

PD: Propagated delay

IDD: Independent departure delay

IAD: Independent arrival delay

TDD: Total departure delay

TAD: Total arrival delay

Then the total delay of flight f_2 comprises of two parts, *propagated delay* and *independent delay*, as introduced by Lan, Clarke and Barnhart [9]. Propagated delay is the delay caused by the late arrival of the previous flight. Independent delay is the delay irrelevant with the previous flight, for example, delay due to weather issues, taxing delays, etc. The equation is given as follows:

$$\begin{aligned} PD \text{ of } f_2 &= \max\{0, \text{delay of } f_1 - \text{Slack}\} \\ IAD \text{ of } f_2 &= TAD \text{ of } f_2 - PD \text{ of } f_2 \end{aligned}$$

In a macro viewpoint of the flight operation, the expected independent delays of all flights can be considered as a constant. Therefore, the objective of the aircraft routing planners is to reduce the total propagated delays.

Now we show an example where robust aircraft routing can make a difference. Assume there are four flight legs. f_1 and f_2 travel from airport 1 to airport 2, and f_3 and f_4 travel from airport 2 to airport 3. The original routing is such that, one aircraft operates f_1 and f_3 , and another aircraft operates f_2 and f_4 (Figure 2). If flight f_1 is delayed by a longer time than its available slack time, in the original routing, f_3 will also be delayed due to propagation, while in the new routing, no flights will be delayed. If flight delay information isn't considered systematically, planners usually adopt the original routing, because it grants more slack time for both the aircrafts.

However, we can see that the new routing is better than the original routing if f_1 often experiences extreme delays. The new routing ensures longer slack time for aircraft 1 to deal with highly possible delays. This example tells that the choice of routing schedules should be made based on the delay information of flight legs.

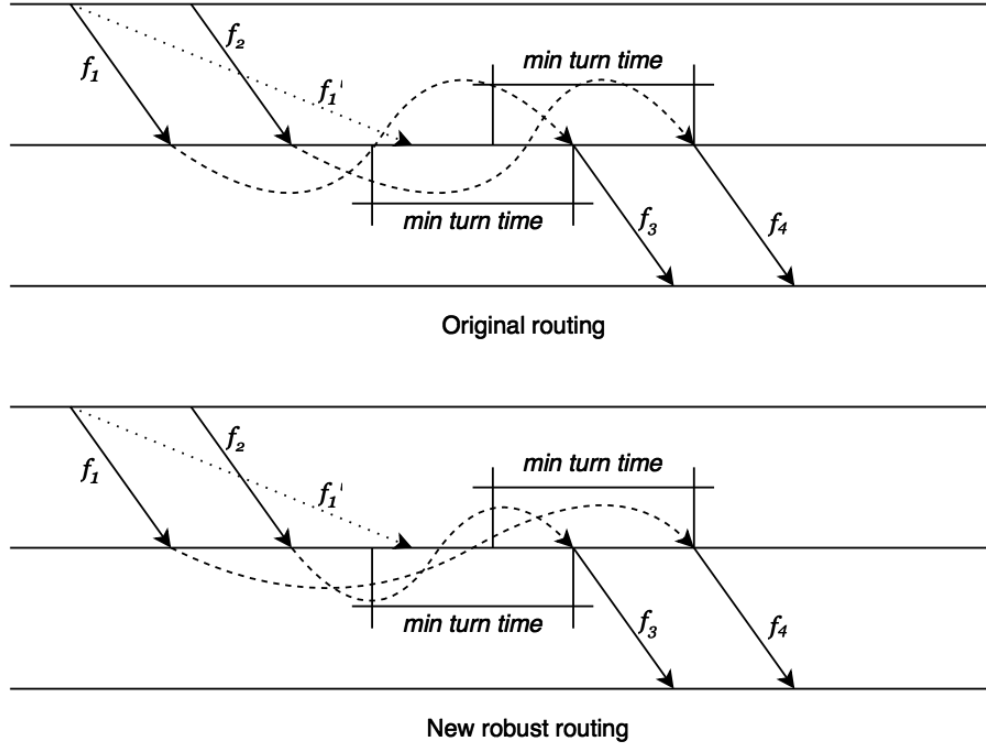


Figure 2 Demonstration of Optimal Slack Allocation

The concept of propagated delays has been discussed in significant detail by Ahmadbeygi et al. [23] and Chirapadhanakul [24], who focus on how robust schedules can be constructed by modeling propagated delays effectively. While this thesis also uses the concept of propagated delays, it differs significantly from existing work by applying the robust models developed by Bertsimas and Sim [19] and Charnes and Cooper [17] and extensions by Marla and Barnhart [16] to this problem.

2.3 Evaluation Metrics

In the field of airline scheduling, there is no single most effective evaluation metric. Moreover, different players in the market tend to consider different metrics. Lan et al. [9], and Marla and Barnhart [16] assess the robustness of routing solutions by three kinds of metrics. (i) Expected on-time performance for all legs in the flight schedule; (ii) Total expected number of passenger disruptions; (iii) Total expected daily flight delay. These metrics are commonly accepted by researchers, and the three can amend each other, and build up a comprehensive set of evaluation metrics.

(i) On-Time Performance

On-Time Performance, such as 15-minute On-Time Performance (15-OTP), measures the percentage of flights that arrive no later than a specific number of minutes (15 minutes) after the scheduled time, which is indicated in the Computerized Reservation System (CRS). It is commonly used by airlines and governments (US DoT) to evaluate airline performance. For the purpose of catching most important delay information, this thesis uses expected on-time performance for all legs in the flight schedule for 15 minutes, 30 minutes, and 90 minutes.

However, merely using On-Time Performance can't evaluate overall performance of airline. The reasons are:

- It does not provide any information about the distribution of delays. Two airlines having the same 15-OTP can have different average delay.

- It does not consider the occurrence of propagated delays in the network. Late Arrival Delay is responsible for a high proportion of delays. Planners need to give larger slack time to the network so that it is robust against delay propagation.
- It does not consider passenger delays. The passenger delays, and passenger missed connections, are considered to be important by both airlines and passengers. If a flight is 10 minutes late, which is not counted in 15-OTP, but it causes a missed connection, the case still needs to be considered as a negative effect of delays.

Therefore, we have the other two metrics in the thesis.

(ii) Total Expected Flight Delay

As Section 2.2 states, total delay of flights is comprised of propagated delay and independent delay. Independent delay can be regarded as a constant when we evaluate the performance, thus total expected flight delay is of concern to us.

Given a routing of an aircraft, mathematically we can calculate the propagated delay and independent delay as follows.

$$\begin{aligned}
 TDD &= \max\{ADT - PDT, 0\} \\
 TAD &= \max\{AAT - PAT, 0\} \\
 Slack_{ij} &= PDT_j - PAT_i - \text{min turn time} \\
 PD_{ij} &= \max\{TAD_i - Slack_{ij}, 0\} \\
 IDD_j &= TDD_j - PD_{ij} \\
 IAD_j &= TAD_j - PD_{ij}
 \end{aligned}$$

(iii) Total Expected Number of Passenger Disruptions

A passenger's itinerary is called *disrupted* if one or more flights in his/her schedule are cancelled, or the connection time of some pair of consecutive flights is not enough for him/her to catch the second flight in the pair. The impact of passenger disruptions is often underestimated because the passenger has to wait for a long time before he/she takes an alternative flight. The number of hours delayed for each such case can be large. Moreover, passenger disruptions cause the airline company to react manually for each individual case. The airline employees need to re-accommodate the disrupted passenger. The average delay for passengers on cancelled flights can be large. Consequently, it is important for airlines to consider passenger delays and disruptions.

CHAPTER 3 ROBUST AIRCRAFT ROUTING MODELS

3.1 The Deterministic Aircraft Routing Model

We start by introducing the standard deterministic aircraft routing formulation, denoted as *AR*. The objective of this model is typically to find a feasible aircraft routing solution such that all aircraft can be subject to mandatory periodic maintenance checks.

The *AR* formulation is set up on a timeline network. This network is similar to a time-space network where each node is a point in space and time. Each node represents either the start point of a flight at the origin airport at the scheduled departure time, or the end point of a flight at the destination airport at the scheduled arrival time. Arcs in this network are divided into flight arcs and ground arcs. Flight arcs connect the starting point of a flight at the origin airport and scheduled departure time with its ending point at the destination airport and scheduled arrival time. Ground arcs connect nodes that are at the same airport and succeed each other in time, to capture aircraft waiting on the ground at a particular airport. Thus, flight arcs represent a flight, and ground arcs represent the period when the aircraft is on the ground. The timeline network spans the maximum time between mandatory maintenance checks of aircraft, which is typically 72 hours.

Figure 3 shows a timeline network with 4 airports. Each solid arrow represents a flight arc, and each dotted arrow represents a ground arc.

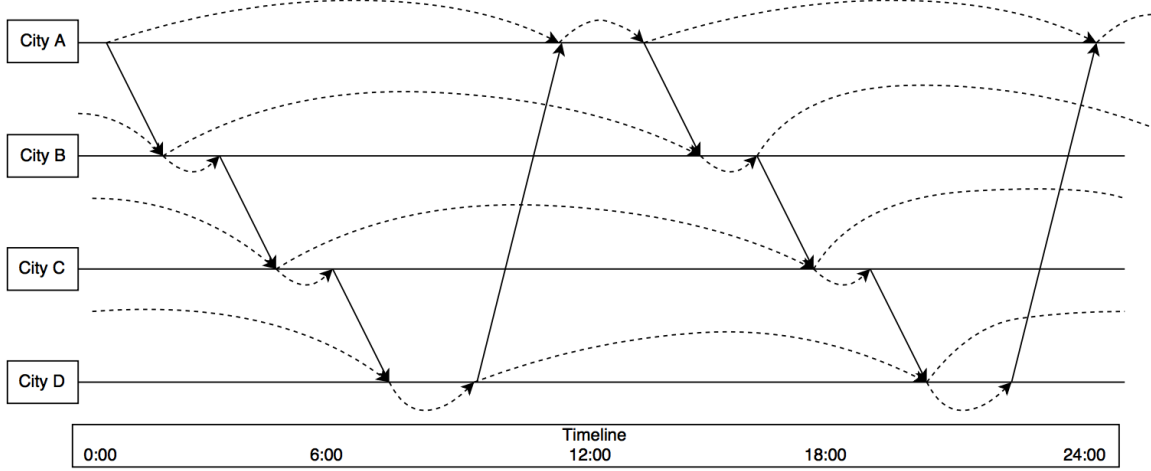


Figure 3 Illustration of a Timeline Network

The decision variables in this formulation are modeled on composite variables called strings. Composite variables capture multiple decisions simultaneously, such that they can be modeled using easier constraints and can result in formulations with structures that are easier to solve. In the context of aircraft routing, each string is a sequence of flights, beginning at a maintenance station (airport where maintenance can be performed) and ending at a maintenance station, operated by a single aircraft, and followed by maintenance at the destination of the final flight.

We now present the standard formulation for aircraft maintenance routing [9]. We first introduce some set notation. Let F be the set of all daily flights, F^+ be the set of flight legs which originate at a maintenance station, and F^- be the set of flight legs which end at a maintenance station. Let S be the set of all possible strings (aircraft routes). The set of ground arcs is denoted as G . The set of flight legs beginning with flight leg i is denoted by S_i^+ , and the set of flight legs ending with flight leg i is denoted by S_i^- .

Second, there are two sets of decision variables. For each string $s \in S$, $x_s = 1$ if string s is selected in the aircraft routing; and 0 otherwise. For each ground arc $g \in G$, y_g is the number of aircraft on g . Some special notations of y_g are included in Constraint (3.3) and (3.4). These are as follows. Variable $y_{i,d}^-$ represents the number of aircraft on the ground just before flight leg i departs and $y_{i,d}^+$ is the number of aircraft on the ground just after flight leg i departs, for all flight legs i . Similarly, $y_{i,a}^-$ is the number of aircraft on the ground just before flight leg i arrives and $y_{i,a}^+$ is the number of aircraft on the ground just after flight leg i arrives, for all flight legs i .

Third, we specify the parameters in this formulation. a_{is} is the cover parameter. a_{is} is 1 if flight leg $i \in F$ is contained in string $s \in S$ and 0 otherwise. We also use in this formulation the concept of count line, which is a particular timestamp, at the same time on each day in the timeline network, to count the total number of aircraft. The count line is a specific time point on the timeline network, for example, midnight on each day. A string can cross the count line multiple times, depending on its length, because the timeline network is multiple days long. r_s is the number of times each string s crosses the count line, p_g is the number of times ground arc g crosses the count line, and N is the number of aircraft available. By setting the length of the timeline network to the time period between maintenance checks, and by setting each string to end with maintenance time, we ensure that by construction, each aircraft is maintained at least once in that time period.

The formulation of the basic aircraft maintenance routing problem, denoted AR, is as follows.

AR:

$$\min 0 \quad (3.1)$$

$$\text{s.t. } \sum_{s \in S} a_{is} x_s = 1 \quad \forall i \in F \quad (3.2)$$

$$\sum_{s \in S_i^+} x_s - y_{i,d}^- + y_{i,d}^+ = 0 \quad \forall i \in F^+ \quad (3.3)$$

$$-\sum_{s \in S_i^-} x_s - y_{i,a}^- + y_{i,a}^+ = 0 \quad \forall i \in F^- \quad (3.4)$$

$$\sum_{s \in S} r_s x_s + \sum_{g \in G} p_g y_g \leq N \quad (3.5)$$

$$y_g \geq 0 \quad \forall g \in G \quad (3.6)$$

$$x_s \in \{0,1\} \quad \forall s \in S \quad (3.7)$$

Expression (3.1) is the objective function. In its basic form, the aircraft routing problem is typically a feasibility problem, therefore AR has objective zero. The AR model does not have a specified objective function, so a solver returns any one of the feasible aircraft routing solutions. Constraint (3.2) ensures each flight leg is operated exactly once, so it is called the *cover* constraint. Constraints (3.3) and (3.4) ensure aircraft *flow balance*, that is, when flights depart from a node or arrive at a node, the total number of aircraft entering the node and the total number of aircraft leaving the node is equal. This is done by ensuring that the flow of aircraft into a node through the incoming flight and ground arcs is the same as the flow of aircraft outgoing from the node through the outgoing flight and ground arcs. Constraint (3.5) ensures that the number of aircraft utilized is constrained by N , which is the available number of aircraft of that fleet type. Because flow balance is already ensured, it is sufficient to ensure that the number of aircraft at a specific point in time, specifically, at the count line, is constrained by N . At the count line, we ‘count’ the total number of aircraft using the flight arcs and the ground arcs that intersect the count line. We refer to this constraint as the *count* constraint. Constraint (3.6) and (3.7) ensure positive values for the y variables and binary values for the x

variables. While both y and x are required to be integer, the integrality of the y variables can be relaxed because constraints (3.3) and (3.4) will ensure that if x is integer, y will also be integer.

3.2 Lan, Clarke, Barnhart's Approach

To make the aircraft routing solution robust to delays, Lan, Clarke, Barnhart [9] attempt to generate robust solutions by considering total expected propagated delay in the objective. As is explained in Section 2.2, the difference between total expected delays and total expected propagated delays is total independent delays, which is a constant in a data-driven approach. Therefore, minimizing total expected propagated delay is equivalent to minimizing total expected delays.

Assuming flight j immediately follows flight i in string s , and the propagated delay between flight leg i and j in s is pd_{ij}^s . Then the total expected propagated delay is written as follows.

$$E \left[\sum_{s \in S} \left(\sum_{(i,j) \in s} pd_{ij}^s \right) x_s \right] = \sum_{s \in S} \left(x_s \sum_{(i,j) \in s} E[pd_{ij}^s] \right) = \sum_{s \in S} x_s d_s \quad (3.8)$$

where $d_s = \sum_{(i,j) \in s} E[pd_{ij}^s]$.

Lan, Clarke, and Barnhart impute the independent and propagated delays of each flight leg based on the operated strings in the historical data. Having computed the independent delay of each flight, they then compute d_s , the expected propagated delay of each string s . Based on this, they formulate the robust model, which is the

same as *AR*, with an objective of minimizing total expected propagated delay. We denote their robust approach as *LCB*, as follows.

LCB:

$$\min \sum_{s \in S} d_s x_s \quad (3.9)$$

$$\text{s.t. Cover, Balance, Count, and Integrality (3.2) - (3.7)} \quad (3.10)$$

3.3 Chance-Constrained Programming Approach

Chance-Constrained Programming (CCP) specifies that the probability of a constraint being satisfied exceeds a pre-specified threshold probability, considering the fact that the various parameters in the constraint are uncertain. The idea of Chance-Constrained Programming is that the probability that the constraint of the model with uncertain parameters is satisfied must be over a predetermined threshold level.

Applied to the aircraft routing problem, it specifies that the probability that each string is operated without potential risks of disruption, exceeds a certain user-specified probability level α . We will define the ‘potential risk of disruption’ below. Under this framework, the general formulation of a chance-constrained model for the *AR* model is as follows:

$$\max 0 \quad (3.11)$$

$$\text{s.t. } \mathbf{P} \left(\sum_{s \in S} a_{is} x_s = 1 \right) \geq \alpha_i \quad \forall i \in F \quad (3.12)$$

$$\text{Balance, Count, and Integrality (3.3) - (3.7)} \quad (3.13)$$

We define p_{is} as the probability in the historical data, that flight leg i in string s is operated with a certain pre-specified service level, that is, the total delay (independent and propagated delay combined) of i when operated by string s , is below a pre-specified threshold of t minutes. By this definition of p_{is} , the probability p_i of flight i being delayed less than t minutes in the chosen solution is $p_i = \sum_{s \in S} p_{is} x_s$.

The real-world interpretation of this is as follows. In our experiments, to be described in the next chapter, we solve this model with different values of t , at 15, 30 and 90 minutes. At delay levels of 15 minutes, flights incur on-time performance delays; at delay levels of 30 minutes, passengers might risk missing their connecting flight; and at delay levels of 90 minutes, flight cancellations may occur, because when a flight is delayed by 15 or 30 minutes, passengers might have risks in catching the following connection flight; whereas if a flight is delayed by 90 minutes, flight cancellation, and thus flight non-coverage can occur. We then write the Chance-Constrained Programming formulation, denoted as *CCP*, as follows.

CCP

$$\max 0 \quad (3.14)$$

$$\text{s.t. } \sum_{s \in S} a_{is} x_s = 1 \quad \forall i \in F \quad (3.15)$$

$$\sum_{s \in S} p_{is} x_s \geq \alpha_i \quad \forall i \in F \quad (3.16)$$

$$\text{Balance, Count, and Integrality (3.3) - (3.7)} \quad (3.17)$$

Constraints (3.16) are the ‘robustness constraints’. α_i is the protection level, such that the probability that flight leg i has a delay less than t for at least α_i percent of

the time. In other words, the probability that flight i is delayed more than t minutes in the chosen solution should be smaller than or equal to $1 - \alpha_i$.

Similar to AR , this model has a feasibility objective function (minimize zero). However, feasible solutions may not exist if the protection level α_i is chosen such to be ‘too high’, such that no solution exists. Thus, the challenge in CCP is the determination of the ‘right’ values α_i that are high enough to decrease the delays but do not result in infeasibility. In practice, we determine the maximum α_i , which can generate feasible solution by repeated re-solving to find the appropriate α -values. Repeated model execution, however, is not ideal in determining α -values, because of the trial-and-error process involved, as well as because of the loss of tractability arising from re-solving.

We therefore propose the $CCP \min EPD$ (CCP with minimize Expected Propagated Delay objective) model to incorporate the features of CCP models and LCB models, and direct the search towards solutions that satisfy multiple criteria. This model has the objective function of the LCB model, and the constraints of the CCP model. It is expected that with an objective function added, $CCP \min EPD$ model will work better than CCP model. The α_i values are determined in the same method as in CCP model.

CCP min EPD

$$\min d_s x_s \quad (3.18)$$

$$\text{s.t. } \sum_{s \in S} a_{is} x_s = 1 \quad \forall i \in F \quad (3.19)$$

$$\sum_{s \in S} p_{is} x_s \geq \alpha_i \quad \forall i \in F \quad (3.20)$$

$$\text{Balance, Count, and Integrality (3.3) - (3.7)} \quad (3.21)$$

3.4 Extended Chance-Constrained Programming Approach

As discussed in Section 3.3, the CCP model has the limitation that it needs to be resolved multiple times to find the best values of protection levels, by trading off feasibility and robustness; resulting in poor tractability. To overcome these limitations, Marla and Barnhart [16] develop the α -CCP model. In the α -CCP model, the protection levels α_i do not need to be specified in advance. Instead, the protection levels are decision variables in the model.

Various objective functions may be used. One possibility, as shown in the formulation α -CCP-1 (3.22) is to maximize the sum of protection levels of all flights in the network.

α -CCP-1

$$\max \sum_{i \in F} \alpha_i \quad (3.22)$$

$$\text{s.t. } \sum_{s \in S} a_{is} x_s = 1 \quad \forall i \in F \quad (3.23)$$

$$\sum_{s \in S} p_{is} x_s \geq \alpha_i \quad \forall i \in F \quad (3.24)$$

$$\text{Balance, Count, and Integrality (3.3) - (3.7)} \quad (3.25)$$

Another possible objective function α -CCP-2 (3.26) is to maximize the minimum protection level over all flights in the network, which would be written as:

α -CCP-2

$$\max \min_{i \in F} \{\alpha_i\} \quad (3.26)$$

$$\text{s.t. } \sum_{s \in S} a_{is} x_s = 1 \quad \forall i \in F \quad (3.27)$$

$$\sum_{s \in S} p_{is} x_s \geq \alpha_i \quad \forall i \in F \quad (3.28)$$

$$\text{Balance, Count, and Integrality (3.3) - (3.7)} \quad (3.29)$$

The objective function contains the minimum α_i values. Optimization models with this kind of objective function can be linearized, as follows.

$$\max z \quad (3.30)$$

$$\text{s.t. } \sum_{s \in S} a_{is} x_s = 1 \quad \forall i \in F \quad (3.31)$$

$$\sum_{s \in S} p_{is} x_s \geq \alpha_i \quad \forall i \in F \quad (3.32)$$

$$z \leq \alpha_i \quad \forall i \in F \quad (3.33)$$

$$\text{Balance, Count, and Integrality (3.3) - (3.7)} \quad (3.34)$$

3.5 Bertsimas and Sim's Extreme Value Approach

We apply the extreme-value robust optimization approach of Bertsimas and Sim to the aircraft routing problem. The essential idea of the extreme-value approach is to minimize the impact of a certain controlled number of uncertain parameters assuming their worst-case values simultaneously. Bertsimas and Sim use a robustness parameter Γ , to express the number of uncertain parameters that are allowed to simultaneously take on their respective worst-case values. For details about the approach, we refer the reader to Bertsimas and Sim [18][19].

Marla and Barnhart [16] apply this to the aircraft routing problem, as follows. Let $\hat{a}_{is} = -1$ if flight $i \in F$ in string $s \in S$ has extreme value of delay exceeding t minutes. Then if flight i is in string s , and the delay exceeds the threshold t in the extreme case (that is, in even one instance in the historical data), it will result in $\hat{a}_{is} = -1$, and thus $a_{is} + \hat{a}_{is} = 0$, so flight i is not covered at the required service level by that string s , in the worst-case. Similar to the *CCP* approach, t can be set to 15, 30, or 90 minutes, depending on the kind of delay or disruption we would like to capture.

We use a set of ‘robustness’ or ‘protection’ parameters Γ_i , for each flight leg i . For each flight $i \in F$, Γ_i represents the number of strings in which flight i cannot experience delays greater than t minutes in any extreme case. The extreme value (*EV*) formulation developed by using the robust optimization approach of Bertsimas and Sim is as follows.

EV

$$\min 0 \tag{3.35}$$

$$\text{s.t. } \sum_{s \in S} a_{is} x_s + \max \left\{ \sum_{s \in S} \hat{a}_{is} x_s, -\Gamma_i \right\} = 1 \quad \forall i \in F \tag{3.36}$$

$$\text{Balance, Count, and Integrality (3.3) - (3.7)} \tag{3.37}$$

The second term in Constraint (3.36) represents the protection level. If $\sum_{s \in S} \hat{a}_{is} x_s \geq -\Gamma_i$, it ensures that each flight i is covered by at least one string that doesn't have extreme delays. If $\sum_{s \in S} \hat{a}_{is} x_s < -\Gamma_i$, it ensures that each flight i is protected against Γ_i extreme-delay situations. The constraint (3.36) can be easily linearized as follows.

$$\sum_{s \in S} a_{is} x_s + u_i = 1 \quad \forall i \in F \quad (3.38)$$

$$u_i \geq \sum_{s \in S} \hat{a}_{is} x_s \quad \forall i \in F \quad (3.39)$$

$$u_i \geq -\Gamma_i \quad \forall i \in F \quad (3.40)$$

Because the formulation seeks to protect against extreme cases, some flights may be present in multiple strings to ensure that the worst-case is not violated in at least one string, for each flight. The solution given by *EV* formulation might not be a practical solution for the aircraft routing problem, but it can be a reference to other formulations.

3.6 Delta Extreme Value Approach

Similar to the *CCP* approach, the *EV* formulation requires multiple executions with different values of parameters Γ_i , because the ‘best’ level of protection available cannot be ascertained *a priori*. To avoid the need to repeatedly solve *EV* models, Marla and Barnhart [16] propose an alternative method, denoted Δ -*EV*. Instead of setting the values Γ_i *a priori*, Γ_i are set as variables and the sum of coverages of all flights in the worst-case is maximized. The objective of Δ -*EV* is to minimize the number of the total number of flight legs that experience extreme delays, provided that each flight leg is covered at least once. The formulation is as follows.

$$\max \sum_{i \in F} \Gamma_i \quad (3.41)$$

$$\text{s.t. } \sum_{s \in S} a_{is} x_s + u_i = 1 \quad \forall i \in F \quad (3.42)$$

$$u_i \geq \sum_{s \in S} \hat{a}_{is} x_s \quad \forall i \in F \quad (3.43)$$

$$u_i \geq -\Gamma_i \quad \forall i \in F \quad (3.44)$$

$$\text{Balance, Count, and Integrality (3.3) - (3.7)} \quad (3.45)$$

In the special structure of this formulation, maximizing $\sum_{i \in F} \Gamma_i$ is equivalent to minimizing $-\sum_{i \in F} \Gamma_i$, and thus minimizing $\sum_{i \in F} u_i$. Because the upper bound of $\sum_{s \in S} \hat{a}_{is} x_s$ is u_i , equivalently, we can minimize $\sum_{i \in F} \sum_{s \in S} \hat{a}_{is} x_s$. Therefore, the simplified Δ -EV formulation is as follows:

Δ -EV

$$\min \sum_{i \in F} \sum_{s \in S} \hat{a}_{is} x_s \quad (3.46)$$

$$\text{s.t. } \sum_{s \in S} a_{is} x_s \geq 1 \quad \forall i \in F \quad (3.47)$$

$$\text{Balance, Count, and Integrality (3.3) - (3.7)} \quad (3.48)$$

3.7 Delta Objective Extreme Value Approach

Because the extreme value robust optimization framework also allows uncertainty to be modeled in the objective function, we present an alternative extreme value formulation, denoted Δ -Obj-EV. The idea of Δ -Obj-EV is to protect against the scenario when certain number (I) of strings in the solution simultaneously experience extreme value of propagated delays, while the other strings in the solution experience no uncertainty, that is, have a propagated delay value of zero.

We present a parameter - maximum propagated delay D , such that D exceeds the sum of the extreme (worst-case) propagated delays of any subset of Γ strings. While this particular scenario is hardly realized in practice, it serves the purpose of choosing strings that have some slack in their propagated delays relative to the threshold D .

As Marla and Barnhart [16] note in their work regarding the a priori specification of the protection parameter Γ , it is more intuitive to allow the formulation to maximize the level of protection within a pre-specified threshold of delay. The Δ -Obj-EV formulation allows the largest number of strings to realize their worst-case propagated delays without exceeding the maximum propagated delay threshold D . This model is presented as follows.

Δ -Obj-EV

$$\min \Delta \tag{3.49}$$

$$\text{s.t. } \sum_{s \in S} \hat{d}_s x_s - \sum_{s \in S} \hat{d}_s v_s \leq D \tag{3.50}$$

$$\Delta \geq \sum_{s \in S} v_s \tag{3.51}$$

$$v_s \leq x_s \quad \forall s \in \bar{S} \tag{3.52}$$

$$v_s \leq w_s \quad \forall s \in \bar{S} \tag{3.53}$$

$$v_s \geq x_s + w_s - 1 \quad \forall s \in \bar{S} \tag{3.54}$$

$$w_s \geq w_{s+1} \quad \forall s \in |S| - |\bar{S}| + 1, \dots, |S| - 1 \tag{3.55}$$

$$w_{|S| - |\bar{S}| + 1} \leq 1 \tag{3.56}$$

$$w_{|S|} \geq 0 \tag{3.57}$$

$$\text{Cover, Balance, Count, and Integrality (3.2) - (3.7)} \tag{3.58}$$

$$v_s \in [0, 1] \quad \forall s \in S \tag{3.59}$$

$$w_s \in \{0, 1\} \quad \forall s \in S \tag{3.60}$$

D is a threshold on total propagated delay, set by examining historical data and choosing a reasonable value that represents a low level of propagated delay in the network. We set the nominal propagated delay value for any string s to be zero, and let \hat{d}_s represent the extreme or worst-case propagated delay value for string s in the historic data. Let \bar{S} be the set of strings $s \in S$ with non-zero extreme values of propagated delay in the historic data, which means, $\hat{d}_s > 0$. v_s is a set of intermediate variables. $v_s = 1$ if string s takes its nominal propagated delay value zero, and $v_s = 0$ if it takes its worst-case value. To maximize the size of the minimal subset of strings that realize their worst-case values, we sort the strings by increasing order of their \hat{d}_s values, such that $\hat{d}_1 \leq \hat{d}_2 \leq \dots \leq \hat{d}_{|\bar{S}|}$. Δ is the maximum number of strings with propagated delays assumed to be nominal. Therefore, by minimizing Δ , we are allowing the maximum number of strings to assume their worst-case values within the allowable threshold D .

Objective function (3.49) minimizes the value Δ , which counts the number of strings in the solution that cannot be protected against realizing their worst-case (extreme) propagated delay values. Constraint (3.50) requires that the total worst-case propagated delay, when $|\bar{S}| - \Delta$ strings realize worst-case delay values, is less or equal to D . The number of non-zero v_s values, which is equal to the number of strings not protected against the realization of their extreme values of propagate delay, is limited by Δ . Constraints (3.52) force $v_s = 0$, unless string s is in the solution. Constraints (3.53) introduce another intermediate variable w_s , such that $w_s = 1$ for string $s \in \bar{S}$, if there exists a $k \geq s$ such that $v_k = 1$. $v_s = 1$ only if $w_s = 1$. Constraints (3.54) allow $v_s = 1$ only if both $w_s = 1$ and $x_s = 1$. These constraints jointly ensure

that maximum number of strings can assume nominal value by forcing those strings with smallest \hat{d}_s values to have $v_s = 1$ if $x_s = 1$. Constraints (3.55) – (3.56) ensure that w_s is decreasingly ordered. Along with the binary constraints and the \hat{d}_s parameters, it makes the maximal set of w_s be set to 1. Constraints (3.59) and (3.60) specify that v_s and w_s are binary variables. It is enough to specify that w_s is continuous between zero and one to ensure that w_s takes on binary values.

CHAPTER 4 EVALUATION AND RESULT ANALYSIS

In this chapter, we discuss the results from the application of the models described in Chapter 3 to real-world data, and the insights obtained from our experiments.

4.1 Experimental Set-up

4.1.1 Description of the Raw Data Sources

To understand the performance of the models presented in Chapter 3, we conduct experiments on the network data of a major airline in the United States. The airline operates a hub-and-spoke network with three major hubs. We obtain the historical schedule, and flight leg delay and cancelation data from the Airline On-Time Performance (AOTP) database, made available by the Bureau of Transportation Statistics (BTS) [25].

The ASQP database records the planned and operated schedule of each flight operated in the US. The fields in this database that are relevant to this work are: the basic flight information in the ‘Date’, ‘Origin’, ‘Destination’, ‘Carrier’, and ‘Tail number’ fields, on-time performance in the ‘Planned departure time’, ‘Actual departure time’, ‘Wheels off time’, ‘Wheels on time’, ‘Planned arrival time’, ‘Actual arrival time’, and ‘Reason of delay if exists’ fields, and disruption information in the ‘Cancelled’, ‘Diverted’, ‘Reasons for cancelation or diversion’ fields.

4.1.2 Pre-processing of the Data

Step 1: Inferring routings based on tail numbers: We create a database of the relevant raw data for the airline of interest. We then separate the flights in the data based on fleet type, filling in information for canceled flights based on the original planned fleet type. This is because the aircraft routing problem is solved separately for flights of each fleet type. We then sequence the flights based on the planned routing, that is, the sequence of flights operated by each aircraft (by its tail number); in order to compute the independent and propagated delays.

Step 2: Finding independent delays: The total delay of each flight is separated into independent and propagated delay based on the procedure described in Lan, Clarke and Barnhart [9]. These equations have been described in detail in Section 2.3. By this procedure, we find independent delay for each flight in the historical data, over various days in the data.

Step 3: Generating fleet sub-networks: Once the independent delays of all flights have been determined, we divide the flight networks into subnetworks that are all operated by the same aircraft type (for example, B737-824 would be one aircraft type). Among these networks, we focus on those that are operated as daily networks, that is, those which are operated more than 200 times in a year (on all days of the week, excluding weekends). We also ensure that these networks are balanced, that is, the number of incoming flights is equal to the number of outgoing flights for each location. We select two such networks for our experiments.

4.1.3 Description of Selected Networks

We provide descriptive statistics for the two selected networks in Table 3. Both are networks where the schedule is repeated daily. Network N_1 has 50 flights and network N_2 has 165 flights. We then set up the timeline network corresponding to networks N_1 and N_2 , that spans four days, because the maintenance period is assumed to be 72 hours. 72 hours must be covered within a 4-day span, because in the worst case, the first flight starts at the end of day 1 and the last flight ends at the beginning of day 4, and thus a 3-day span cannot cover the maintenance period. We then generate the strings in each network that are of length less than or equal to 72 hours. Recall that by definition, each string is a series of flights, beginning and ending at a maintenance station, followed by maintenance at the destination maintenance station. Therefore, we ensure that each aircraft is available to depart before the end of the fourth day. Upon enumeration, we see that there are 9,639 strings in N_1 and 878,207 strings in N_2 . The number of available aircraft is 20 for the first network, and 61 for the second.

We then evaluate the performance of each of the models described in Chapter 3 on all the scenarios for each network. For N_1 , because of limited number of scenarios available, we fit the historical independent delay values to a distribution, and these were seen to best fit a lognormal distribution. We then sampled from this distribution to generate 5000 scenarios for the daily network. The specifications can be found in Appendix A.1. For N_2 , there are a significant number of scenarios available in the historical data, that is, 200 realized scenarios of 365 days in the year are available. Therefore, we directly use these independent delay data as scenarios in our experiments.

Network No.	Daily Flights	Aircraft	Strings	Scenarios
N_1	50	20	9,639	5000
N_2	165	61	878,207	200

Table 3 Summary of network characteristics

4.1.4 Generating Strings in each Network

We generate a timeline network for each network in our experiments and enumerate strings using a depth-first search process. The process of enumerating the strings automatically incorporates feasibility constraints, by definition. Each string consists of a sequence of flight legs beginning and ending at a maintenance station, with minimum turn time between successive legs, and the last flight followed by maintenance. For any two successive flights, the destination airport of the first flight is the same as the origin airport of the second, with turn time for the aircraft between the legs at least as large as the minimum turn time. The duration of each string is at least two days (to eliminate too-short routes) and less than 72 hours, thus satisfying the constraint of maintaining the aircraft every 72 hours.

4.1.5 Generating Inputs to Mathematical Models

We introduce the idea of *scenario* here. In the historical data, each flight leg is operated several times, and we compute its independent delay value as described above. We regard each independent delay value (for each day) as one scenario for this flight leg. In our data-driven approach, we assume the independent delay of the flight legs has the same distribution as the historical independent delay values. In

the case of network N_2 , we have sufficient daily data of the independent delays in the year of historical data considered. Therefore we use that data as the scenarios in our experiments. In the case of network N_1 , we have a limited number of scenarios where all flights are operated. The number of scenarios is sufficient to fit the data to distributions but fall short for testing the model performance. Therefore, after fitting the distributions, we simulate 5000 scenarios to capture the distributions. We assume that the delay distributions of different flights' independent delays are statistically independent of each other for N_1 but not so for N_2 .

We then compute the propagated delay of strings over the various scenarios. In doing so, because the strings are of length greater than a day, we make sure to consider the correct value of independent delay corresponding to that scenario. Therefore, for each string beginning on a certain day (scenario), we compute the propagated delay of the entire string by the method described in Chapter 2. We thus have the propagated delay of each string for all scenarios.

From this data, we generate the inputs to these models, such as the matrix for Cover constraint parameters, the matrix for Balance constraint parameters, the matrix for Count constraint parameters, and vectors for statistics of propagated delay data.

4.1.6 Running Robust Aircraft Routing Models

The final step of our experimental procedure is to run all the models written in Chapter 3. We solve our models in Java, integrated with *IBM ILog CPLEX v12.5.1*.

In running our models, we assume that there will not be recovery interventions such as flight cancellation or aircraft swaps, but allow the delays to be propagated along the strings. In this way, we can estimate the robustness of solutions before intervention. Because cancelation and swap strategies vary among different airlines, this would enable us to analyze solution robustness in a fair manner across all airlines.

4.1.7 K-Fold Cross Validation

Statistically, overfitting occurs when a statistical model describes random error or noise instead of the underlying relationship. Overfitting generally occurs when a model is excessively complex, and fits to the available data too well, however, when the same model is used on a different data set, it proves to be poor in explaining the phenomena or in predicting the outcomes of those observations. If we take a different sample of validation data from the training data, it might turn out that the solution doesn't fit the validation data as well as the training data.

To overcome overfitting, we use a k -Fold Cross Validation approach for the various models discussed in Chapter 3 (AR , LCB , CCP , α - CCP , EV , Δ - EV , Δ - Obj - EV). We partition the sample data randomly into k equal-size subgroups. From the k subgroups, $k-1$ subgroups are used as training data to fit the parameters of the models described in Chapter 3. Once the models are solved, the k^{th} subgroup is used as validation data, to evaluate the performance of the solutions of the models. The cross validation process is executed k times, such that each of the k subgroups is treated as validation data exactly once. The k results thus obtained are then averaged for analysis. In our experiments, we set $k = 5$.

4.2 Analysis of Results

4.2.1 Computation Time

Table 4 reports the average computation times for the network N_2 . For network N_1 , the computational times are usually under 1 second, so it cannot accurately represent the computational complexity of the models.

Model	Parameters	Iterations	Run time per iteration (sec)
AR	None	1	25.14
LCB	None	1	42.81
CCP	$\alpha_i \forall i$	1 for each α	52.30
α -CCP-1	None	1	72.91
α -CCP-2	None	1	60.61
EV	$\Gamma_i \forall i$	1 for each Γ	45.33
Δ -EV	None	1	42.13
Δ -Obj-EV	1	1	6018.39

Table 4 Complexity and Run Times

For the CCP and EV models, multiple iterations are required to determine the appropriate α and Γ values. So the run time is calculated as the average of several runs of the same type of model.

According to the run times in Table 3, we can see that all the models except for Δ -Obj-EV model are all at the same level of complexity. Although α -CCP and Δ -EV

tend to have more specifications in constraints and objective function, they experience the same level of run times with *CCP* and *EV*, respectively. On the other hand, Δ -*Obj-EV* has a very high run time, due to the large number of strings. It is worth some caution if we need to use it in an even larger network, because the number of constraints grows exponentially (same as the number of strings) as the number of flights grows.

4.2.2 Comparison of Solution Quality

In this section, we compare the quality of solutions from all the models in terms of three evaluation metrics: on-time performance, average total propagated delay and passenger disruptions. In particular, for on-time performance, we look at 15-min on-time performance (OTP), 30-min OTP, 60 min OTP, 90 min OTP, 120 min OTP and 180 min OTP. As described in Chapter 3, 15-min OTP relates to whether a flight is denoted ‘late’ by the Bureau of Transportation Statistics, the 30-min OTP to whether passengers can make their connecting flights, the 60-minute OTP to whether crew can make connections; and the 90-min, 120-min and 180-min OTP to policies of potential flight cancelations. We use the 5-fold cross validation method to generate 5 sets of evaluation values, and then we average them to eliminate the overfitting effect, and have a fair comparison.

Table 5 and Table 6 are the on-time performance, total propagated delay and passenger disruptions metrics for network N_1 . For N_1 , we notice that α -*CCP-1* and Δ -*EV* model perform the best in terms of on-time performance and disrupted passengers, and the propagated delays are also close to the best. *LCB*, *CCP min EPD* and α -*CCP-2* perform second best in terms of our metrics of interest. *CCP*, *EV* and Δ -

Obj-EV perform rather poorly compared to the other models. Nevertheless, they are still better than the routing used by the airline.

	Flight On-Time Performance					
	≤ 15 min	≤ 30 min	≤ 60 min	≤ 90 min	≤ 120 min	≤ 180 min
<i>Airline's Routing</i>	82.61%	91.21%	96.37%	98.00%	98.80%	99.44%
<i>LCB</i>	89.63%	95.39%	98.36%	99.18%	99.55%	99.82%
<i>CCP</i>	84.36%	92.71%	97.24%	98.56%	99.16%	99.63%
<i>CCP Min EPD</i>	89.60%	95.37%	98.35%	99.18%	99.55%	99.82%
α -CCP-1	90.89%	95.87%	98.50%	99.24%	99.59%	99.83%
α -CCP-2	89.83%	95.32%	98.26%	99.10%	99.49%	99.78%
<i>EV</i>	83.68%	93.04%	97.37%	98.61%	99.19%	99.63%
Δ -EV	90.89%	95.87%	98.50%	99.24%	99.59%	99.83%
Δ -Obj-EV	82.76%	91.42%	96.47%	98.06%	98.84%	99.46%

Table 5 On Time Performance Results for Network N_1

	Propagated Delay	Passenger Disruptions	
	Total Average	# Disrupted Passengers	Percentage
<i>Airline's Routing</i>	95.90	109	6.50%
<i>LCB</i>	61.88	74	4.40%
<i>CCP</i>	81.35	96	5.74%
<i>CCP Min EPD</i>	61.87	74	4.39%
α -CCP-1	62.35	66	3.94%
α -CCP-2	65.51	74	4.41%
<i>EV</i>	107.06	78	4.68%
Δ -EV	61.43	66	3.94%
Δ -Obj-EV	93.85	108	6.46%

Table 6 Propagated Delay and Passenger Disruption Results for Network N_1

Table 7 and Table 8 present the same set of performance metrics for network N_2 . In N_2 , we notice that *CCP min EPD* performs best across all the three metrics. *LCB*,

α -CCP-1 and Δ -EV continue to perform very well. By contrast with N_1 , where Δ -Obj-EV does quite badly, in N_2 , the performance of Δ -Obj-EV is reasonable and can be regarded as a second-best solution.

	Flight On-Time Performance					
	≤ 15 min	≤ 30 min	≤ 60 min	≤ 90 min	≤ 120 min	≤ 180 min
<i>Airline's Routing</i>	72.92%	81.40%	89.45%	93.66%	96.23%	98.58%
<i>LCB</i>	73.43%	81.92%	89.90%	93.95%	96.46%	98.63%
<i>CCP</i>	73.61%	82.01%	89.90%	93.91%	96.37%	98.60%
<i>CCP Min EPD</i>	74.55%	82.82%	90.41%	94.27%	96.63%	98.68%
α -CCP-1	73.79%	82.07%	89.96%	93.96%	96.44%	98.62%
α -CCP-2	73.16%	81.69%	89.68%	93.76%	96.30%	98.58%
<i>EV</i>	72.73%	81.33%	89.48%	93.66%	96.25%	98.58%
Δ -EV	73.54%	81.80%	89.75%	93.83%	96.31%	98.55%
Δ -Obj-EV	73.55%	81.98%	89.90%	93.92%	96.43%	98.62%

Table 7 On Time Performance Results for Network N_2

	Propagated Delay	Passenger Disruptions	
	Total Average	# Disrupted Passengers	Percentage
<i>Airline's Routing</i>	458.59	870922	6.35%
<i>LCB</i>	412.77	73149	5.33%
<i>CCP</i>	342.26	67170	4.89%
<i>CCP Min EPD</i>	342.56	71241	5.19%
α -CCP-1	394.61	68798	5.01%
α -CCP-2	437.61	76487	5.57%
<i>EV</i>	443.18	88562	6.45%
Δ -EV	516.15	87695	6.39%
Δ -Obj-EV	371.25	69185	5.04%

Table 8 Propagated Delay and Passenger Disruption Results for Network N_2

For both the networks, we notice that the α -CCP solution is better than the CCP solution, and the Δ -EV solution is better than the EV solution. This verifies our expectations, because these models were intended to maximize protection level of the solutions. We discuss the reason behind our observations in the next section.

4.2.3 Comparison of Model Performances

According to the comparisons in Section 4.2.2, we synthesize the performances of the models in Table 9.

Tier	Models	Comments
1	<i>LCB, CCP min EPD, α-CCP-1, Δ-EV</i>	Recommend
2	<i>Δ-Obj-EV</i>	Neutral
3	<i>CCP, EV, α-CCP-2</i>	Not Recommend

Table 9 Levels of Recommendation

The models designated as ‘Tier 1’ perform consistently well in the two networks and across the variety of performance criteria that we considered. *LCB* has the objective of minimum total propagated delay. This tends to decrease propagation between successive flights, resulting in high correlation with better on-time performance and lower passenger disruption rate. *CCP min EPD* performs the same or better than *LCB* in our experiments. Because it constrains the probability of delay of each flight, and has the objective of minimizing propagated delay, it can explicitly control flight on-time performance as well as propagated delay. The fact that it incorporates two modeling paradigms facilitates its superior performance. α -CCP-1 improves CCP

models by considering the overall summation of protection levels. This leads to a better on-time performance. Δ -EV improves the EV model by adding an objective function that minimizes the total number of potentially delayed flights. In addition to the definition of \hat{a}_{is} that defines the level of potential delay, the objective function drives the search towards a solution that has the highest protection level possible. Therefore, it works towards higher on-time performance and lower passenger disruption rate. Moreover, all models designated as ‘Tier 1’ have run times that are acceptable for a planning problem. We therefore recommend that airline planners execute these four models in the planning phase to find robust solutions, and choose the best solution using simulation and other customized criteria of interest to the airline.

In Tier 2, consider Δ -Obj-EV. This model performs poorly for N_1 but well for N_2 . This is because the model considers the worst-case propagated delay, but that does not necessarily correlate well with the flight-level delay or propagations between flights. This inconsistency decreases the reliability of this model. The constraints of this model are related to worst-case propagated delay, but not directly to any of the metrics of interest. Moreover, the level of complexity of Δ -Obj-EV is quite high, due to which it might not be solvable in limited time for a larger network. So this would not be superior than the models in Tier 1.

We categorize CCP, EV and α -CCP-2 as ‘Tier 3’. These models, even by our theoretical judgment, as described in Chapter 3, do not perform very well because of the trial-and-error nature of CCP and EV, as well as the difficulty in setting the parameters. Therefore, they are dominated by their respective counterparts. Our results strongly concur with this theory. Similarly, α -CCP-2 only focuses on maximizing the

minimum protection level and so does not focus on the individual protection levels. Therefore, we do not recommend these models, because they are strictly dominated theoretically and empirically by the models in Tier 1.

CHAPTER 5 CONCLUSIONS

In this work, we studied the problem of robust aircraft routing. We built upon the concept of propagated delay proposed by Lan, Clarke and Barnhart [9]. To find a robust aircraft routing in terms of on-time performance (at various thresholds) and passenger disruptions, we propose a series of models based on the work of Marla and Barnhart [16]. Primarily they use the Chance-Constrained Programming (*CCP*) approach and the Robust Optimization or Extreme-Value (*EV*) approach of Bertsimas and Sim, and apply them to the aircraft routing problem. They also propose advanced models that overcome the shortcomings of the original Chance-Constrained Programming model and the Robust Optimization approach. The *CCP* model applies a protection level to each flight in the network. α -*CCP* models extend *CCP* by having an objective function that considers overall protection levels. *EV* model specifies that each flight is protected against Γ worst-case occurrences. The Δ -*EV* model extends *EV* by minimizing the number of potentially delayed flights. Additionally, the Δ -*Obj-EV* was proposed with the objective of maximizing the number of flights that can achieve their worst-case propagated delay values within a pre-specified threshold.

We test these models on the aircraft routing problem. Our experiments are conducted on two daily networks, one with 50 flights and the other with 165 flights, over a four-day planning horizon. We use a K -fold cross validation method to avoid the overfitting effect. We solve the models and compare the results according to three metrics – on time performance (at various thresholds), total expected propagated delay, and passenger disruptions. We found that *LCB*, *CCP min Obj*, α -*CCP-1*, Δ -*EV* work best to generate robust solutions for the aircraft routing problem.

We also find that these models perform better because they allow protection levels to be variable, and maximized by the formulation. Moreover, we also find that minimizing average propagated delay also is key in reducing delay propagation from flight to flights, thus constraining total delay.

Further research in the area of robust aircraft routing can be along three different directions. One is to explore other ways of capturing risk in the context of aircraft routing. For example, metrics beyond constraint satisfaction used in *CCP* or the protection parameter in *EV*, such as the conditional-value-at-risk, or kurtosis, might prove more helpful to understand the risk involved with aircraft routings. Another direction of exploration is to examine the properties of strings and explain the associated distribution of delays related to these metrics. Another important aspect is to find the relationship between solution robustness and network characteristics. We can relate the solution metrics to the prerequisites of modeling – the graph features of network, and the statistical distribution of the independent delays. By studying this relationship, we may be able to predict the goodness of the solutions based on the network structure, and recommend appropriately applicable models. The third direction is to compare different robust routing models under recovery strategies. Currently we didn't incorporate recovery strategies but modeling those can potentially give more accurate understanding of how well these models perform relative to each other.

APPENDIX A INDEPENDENT DELAY DATA

A.1 Independent Delay Simulation for the 50-flight network

For the 50-flight network, we randomly generate a series of 5,000 scenarios, by using lognormal distribution, based on the mean value μ and standard deviation σ .

We use the equation as follows.

$$X = e^{\mu + \sigma Z} \quad (A.1)$$

where X is the random variable to be simulated, and Z is a standard normal variable.

The specific mean and standard deviation data are as in Table 10.

Flight Number	μ	σ
1	24.05	40.40
2	8.74	18.74
3	12.53	30.63
4	7.53	17.94
5	23.63	44.88
6	3.21	3.79
7	6.05	17.23
8	5.21	13.81
9	23.05	44.37
10	6.84	8.11
11	6.53	21.47
12	38.26	62.18
13	19.58	64.69
14	0.26	0.93
15	10.11	16.06
16	25.00	49.59

Table 10 Simulation mean and standard deviations for the 50-flight network

17	7.11	13.34
18	13.74	33.58
19	7.63	12.24
20	11.16	23.37
21	10.21	19.77
22	5.53	14.75
23	9.26	15.22
24	11.16	16.95
25	22.47	35.70
26	4.37	8.08
27	16.32	51.06
28	13.42	18.26
29	24.42	65.06
30	1.11	2.83
31	23.26	34.40
32	43.89	96.06
33	4.84	7.43
34	7.16	5.92
35	4.11	6.22
36	6.58	14.90
37	4.26	7.38
38	11.00	21.04
39	28.53	89.19
40	1.42	4.29
41	8.53	16.99
42	3.32	9.27
43	12.95	25.46
44	10.21	9.49
45	5.26	8.16
46	6.68	20.64
47	6.42	6.25
48	2.26	7.96
49	16.42	11.56
50	2.79	6.46

Table 10 (Continued)

Simulation mean and standard deviations for the 50-flight network

REFERENCES

- [1] C. Barnhart, P. Belobaba, and A. R. Odoni, "Applications of Operations Research in the Air Transport Industry," *Transportation Science*, vol. 37, no. 4, pp. 368–391, Nov. 2003.
- [2] M. Ball, C. Barnhart, M. Dresner, K. Neels, A. Odoni, E. Perterson, L. Sherry, A. Trani, B. Zo, "Total Delay Impact Study: A Comprehensive Assessment of the Costs and Impacts of Flight Delay in the United States," 2010. *NEXTOR*. Available: [/http://www.isr.umd.edu/NEXTOR/rep2010.htmlS](http://www.isr.umd.edu/NEXTOR/rep2010.htmlS).
- [3] Airlines For America, "Per-Minute Cost of Delays to U.S. Airlines," 2014. [Online]. Available: <http://airlines.org/data/per-minute-cost-of-delays-to-u-s-airlines/>. [Accessed: 20-Mar-2015].
- [4] Federal Aviation Administration, "Types of Delay," 2013. [Online]. Available: http://aspmhelp.faa.gov/index.php/Types_of_Delay. [Accessed: 19-Mar-2015].
- [5] L. Marla, "Robust Optimization for Network-based Resource Allocation Problems under Uncertainty," pp. 1–132, 2007.
- [6] S. Shebalov and D. Klabjan, "Robust Airline Crew Pairing: Move-up Crews," pp. 1–18, Feb. 2004.
- [7] J. M. Rosenberger, E. L. Johnson, and G. L. Nemhauser, "A Robust Fleet-Assignment Model with Hub Isolation and Short Cycles," *Transportation Science*, vol. 38, no. 3, pp. 357–368, Aug. 2004.
- [8] A. J. Schaefer, E. L. Johnson, A. J. Kleywegt, and G. L. Nemhauser, "Airline Crew Scheduling Under Uncertainty," *Transportation Science*, vol. 39, no. 3, pp. 340–348, Aug. 2005.
- [9] 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*, vol. 40, no. 1, pp. 15–28, Feb. 2006.
- [10] J. W. Yen and J. R. Birge, "A Stochastic Programming Approach to the Airline Crew Scheduling Problem," *Transportation Science*, vol. 40, no. 1, pp. 3–14, Feb. 2006.

- [11] B. C. Smith and E. L. Johnson, "Robust Airline Fleet Assignment: Imposing Station Purity Using Station Decomposition," *Transportation Science*, vol. 40, no. 4, pp. 497–516, Nov. 2006.
- [12] E. K. Burke, P. De Causmaecker, G. De Maere, J. Mulder, M. Paelinck, and G. V. Berghe, "A multi-objective approach for robust airline scheduling," *Computers and Operation Research*, vol. 37, no. 5, pp. 822–832, Oct. 2009.
- [13] E. K. Burke, P. De Causmaecker, and G. De Maere, "Multi-objective approaches for robust airline scheduling (working paper)," pp. 1–41, Feb. 2015.
- [14] O. Weide, D. Ryan, and M. Ehrgott, "An iterative approach to robust and integrated aircraft routing and crew scheduling," *Computers and Operation Research*, vol. 37, no. 5, pp. 833–844, Oct. 2009.
- [15] M. Sohoni, Y.-C. Lee, D. Klabjan, "Robust airline scheduling under block-time uncertainty," 2011. *Transportation Science*. 45, pp. 451-464.
- [16] L. Marla and C. Barnhart, "Robust Optimization: Lessons Learned from Aircraft Routing," 2010. [Working Paper]
- [17] A. Charnes and W. W. Cooper, "Chance Constrained Programming," *Management Science*, vol. 6, no. 1, pp. 73–79, 1959.
- [18] D. Bertsimas and M. Sim, "Robust discrete optimization and network flows," *Mathematical Programming*, vol. 98, no. 1, pp. 49–71, Sep. 2003.
- [19] D. Bertsimas and M. Sim, "The Price of Robustness," *Operations Research*, vol. 52, no. 1, pp. 35–53, Feb. 2004.
- [20] M. Arikan, V. Deshpande, and M. Sohoni, "Building Reliable Air-Travel Infrastructure Using Empirical Data and Stochastic Models of Airline Networks," pp. 1–47, May 2012.
- [21] C. Yan and J. Kung, "Robust Aircraft Routing," pp. 1–28, Nov. 2014.
- [22] D. Bertsimas and A. Thiele, "Robust and Data-Driven Optimization: Modern Decision-Making Under Uncertainty," pp. 1–39, May 2006.
- [23] S. Ahmadbeygi, A. Cohn, and M. Lapp, "Decreasing airline delay propagation by re-allocating scheduled slack," *IIE transactions*, vol. 42, no. 7, pp. 478–489, 2010.
- [24] V. Chiraphadhanakul, "Routing and Scheduling Models for Robust Allocation of Slack," SM MST/OR Thesis, MIT (2010), May 2010.

- [25] Bureau of Transportation Statistics, "On-time performance data," 2015.
[Online]. Available:
http://www.transtats.bts.gov/DL_SelectFields.asp?Table_ID=236
[Accessed: 20-Mar-2015].