PREEMPTIVE REROUTING OF AIRLINE PASSENGERS UNDER UNCERTAIN DELAYS

by

Lindsey Ann McCarty

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Applied and Interdisciplinary Mathematics)
in The University of Michigan
2012

Doctoral Committee:

Associate Professor Amy E. Cohn, Co-chair
Associate Professor Divakar Viswanath, Co-chair
Professor Andreas R. Blass
Professor Anna C. Gilbert
Associate Professor Kristen S. Moore

ACKNOWLEDGEMENTS

First, I would like to thank my advisors, Amy Cohn and Divakar Viswanath. Dr. Viswanath was especially helpful when I was learning C++ for the implementation of my models and also gave valuable advice concerning my job search and career choices. Dr. Cohn was instrumental in my decision to conduct research in airline scheduling, since I became interested in it after taking a linear programming class with her. She has been a caring, supportive, and knowledgeable advisor. All of her students agree that she throws great parties as well!

I also want to acknowledge all of my professors who helped me grow in my appreciation of mathematics, especially Kevin Roper and Darrin Frey from Cedarville University, where I completed my undergraduate studies. Professor Robert Schumacher taught my first operations research course, and his enthusiasm guided me to pursue the topic in my graduate courses. I appreciate the time and support of the other members of my dissertation committee, Andreas Blass, Anna Gilbert, and Kristen Moore. In particular, Dr. Blass and Dr. Gilbert gave helpful comments on the dissertation, and I appreciate Dr. Moore's support during my years at the University of Michigan. Gavin LaRose and Karen Rhea were great sources of advice and encouragement regarding teaching while I was a graduate student instructor. Also, the staff in the mathematics department was helpful, including Tara McQueen, especially with the transition to graduate school and my job search.

I am appreciative of the members of Graduate Christian Fellowship at the University of Michigan and my friends in the mathematics and IOE departments. These

friends have been an integral part of surviving graduate school, providing encouragement and giving perspective on life. My Bible study from GCF has always been refreshing, providing needed breaks from classes and research.

Most of all, I want to thank my family. Devon McCarty has experienced all of the ups and downs of graduate school with me, has been patient through my studies and hard work, and has even shared his knowledge of programming with me. Without his support, I could not have been successful as a Ph.D. student. Also, I have experienced the journey of graduate school with my sister and brother in law, Ashley and Darren Holland, who came to the University of Michigan at the same time as me. My parents, Stephen and Kimberly Selegue, have been dedicated to visiting me and providing needed encouragement. They have always been an inspiration to me, both academically and personally.

TABLE OF CONTENTS

CHAPTER

LIST OF FIGURES

Figure

LIST OF TABLES

Table

LIST OF APPENDICES

Appendix

ABSTRACT

Preemptive Rerouting of Airline Passengers under Uncertain Delays

by

Lindsey Ann McCarty

Chairs: Amy E. Cohn and Divakar Viswanath

An airline's operational disruptions can lead to flight delays that in turn impact passengers, not only through the delay itself but also through possible missed connections. Much research has been done on crew recovery (rescheduling crews after a flight delay or cancellation), but little research has been done on passenger reaccommodation. Our goal is to design ways that passenger reaccommodation can be improved so that passengers can spend less time delayed and miss fewer connections.

Since the length of a delay is often not known in advance, we consider *preemptive rerouting* of airline passengers before the length of the delay is known. Our goal is to reaccommodate passengers proactively as soon as it is known that a flight will be delayed instead of waiting until passengers have missed connections and to use known probabilities for the length of delay. In addition, we consider all of the affected passengers together so that we can effectively handle passengers' competition for available seats. We can give certain seats to people with short connections or those connecting to international flights.

When there is one delayed flight, we model the problem as a two-stage stochastic programming problem, with first-stage decisions that assign passengers initial itineraries and second-stage decisions that re-assign any passengers who are subsequently disrupted by the delay. We present a Benders decomposition approach to solving this problem. Computational results for this model are given, showing its effectiveness for reducing the length of passenger delays.

When there is more than one delayed flight, we define a *portfolio model* which assigns passengers to *portfolios* that define their itineraries under all possible disruption outcomes. We focus on computational methods for solving this model.

CHAPTER I

Introduction

1.1 Description of the Real World Problem

Airlines often face unexpected disruptions of their scheduled flight times. These disruptions are caused by bad weather, late passengers or crew, congestion in the sky or airports, and issues with the aircraft itself, such as maintenance problems. Operational disruptions for airlines can lead to flight delays that in turn impact passengers, not only through the delay itself but also through possible missed connections. Some passengers may miss events at their destinations. Pilots' and cabin crews' schedules are affected, possibly disrupting later flights or violating strict work-time rules. When airlines cancel flights, they may lose some passengers' trust and also have to reaccommodate passengers on other flights.

Little research has been done on passenger reaccommodation, and there is no formal method like there is for other problems such as crew recovery. Reaccommodation is often done on an ad hoc basis, where each passenger is considered separately and only after a connection has been missed. Some airlines may watch for passengers that will be most affected when flights are delayed, especially for those who have flights that are international or are at the end of the day, but reaccommodation is mostly done manually. Airlines often try harder to reaccommodate passengers that paid a higher fare, are in first class seating, or are frequent flyers. The problem of how to better reaccommodate passengers after delays interests many people and is important for the industry.

1.2 Background and Overview of Airline Scheduling

Many of an airline's planning and recovery problems can be represented by linear or non-linear programming problems to be optimized. Airline planning includes the schedule design problem, the fleet assignment problem, the maintenance routing problem, the crew scheduling problem, and revenue management. Schedule design involves choosing what flights to offer in what markets and at what times, fleet assignment refers to assigning a type of aircraft to each flight, maintenance routing refers to choosing what individual aircraft to assign to each flight to satisfy maintenance requirements, and crew scheduling involves assigning crews to flights (*Barnhart and Cohn* (2004)). Revenue management involves deciding how many seats to provide at each fare level (*Belobaba et al.* (2009)).

Much research has been done on the fleet assignment problem. In *Hane et al.* (1995), the authors model the problem as a multi-commodity flow problem with side constraints for one day's worth of flights. In the network, each node represents an available fleet type, a city, and a time of landing or departing, and the commodities are the fleet types. Because of degeneracy and many integer variables, the authors use an interior-point algorithm and strategies for branching, among other things.

According to *Barnhart et al.* (2002), many times when working on FAM (fleet assignment model), researchers consider revenue for flights and also assignment costs, which include the cost to operate a flight, the cost for carrying the passengers, and the spill cost, all of which depend on the fleet type assigned. The spill cost is the revenue lost from assigning an aircraft that is too small to accommodate all the demand for a flight, although some of the passengers may still buy tickets for other flights, called recapture. Most models ignore or approximate spill costs, use average fares, and assume that demand does not change with the day or time of year. In *Barnhart et al.* (2002), the authors create the Itinerary-Based Fleet Assignment Model that more accurately approximates spill costs and recapture of passengers, producing better solutions for fleet assignments. Interestingly, the Passenger Mix Model is also described, where decisions are made of what fraction of passengers from each itinerary to spill to each other itinerary given a solution to FAM. The authors of *Jacobs et al.* (2008) use origin and destination network effects and expected passenger flows and decompose their resulting fleet assignment model to make it easier to solve.

Crew scheduling is another airline planning problem that has been studied in depth. In *Belobaba et al.* (2009), the authors give a thorough overview of both the crew pairing problem and the crew rostering problem. In the crew pairing problem, sets of flights spanning one to five days called pairing are generated. In the crew rostering problem, the pairings are made into longer schedules, such as thirty days, for a certain person. Specifically, a model is given for the crew pairing problem where there is a set of feasible pairings with the constraint that each flight leg must be included in exactly one chosen crew pairing.

In *Kohl and Karisch* (2004), the authors discuss the crew rostering problem, which involves complex laws to follow with the goal of minimizing cost while producing quality schedules for crews. The author presents a model that is a set partitioning problem, where each schedule must be given to exactly one person, with extra constraints based on a person's qualifications, needs, planned absences, rest time, and other factors.

In *Belobaba* (1989), the author explains how airlines enforce limits on the number of seats available on flights at each price in order to gain more profit as part of revenue management. The author discusses the complexity of this problem, since airlines must predict future demands and reservations. Specifically, a probabilistic decision model is given, where the author makes use of a Gaussian probability density function for the stochastic demand for flights. Using probabilities of passengers not purchasing tickets because of pricing and defining the expected marginal revenue for adding one more available seat at a certain price, the author describes “protection levels” used to decide how many seats to make available at each price level. He shows how using a computerized version of the Expected Marginal Seat Revenue decision model improves revenue for a certain airline.

Since many of an airline’s planning problems have a large data set, it is difficult to solve more than one problem at a time, which produces a sub-optimal solution for the entire process. If the problems could be solved together instead of in succession, more options would be available at each step along the way, giving the significant possibility of lower-cost and higher-profit solutions. Some advances have been made in strategically solving two or more problems simultaneously.

In *Rexing* (1997) and *Rexing et al.* (2000), the authors allows small changes in an original flight schedule in order to give better options while solving the fleet assignment problem. A time window is given to each flight and the times are discretized, where the model is based on the basic FAM model. It is a network flow problem, where some variables represent assigning a fleet type to a flight, and other variables represent the number of aircraft on the ground at certain times for specific fleet types. The costs minimized are operating costs and spill costs of not providing itineraries when there is demand for them at a certain price. The first of two models presented is quicker to solve and involves solving the problem directly. The second uses less memory and is iterative, where necessary flight arc copies are added to the model if needed, rather than including all possible flight arc copies from the beginning.

In *Lohatepanont and Barnhart* (2004), the authors solve the schedule design problem and the fleet assignment problem. They present an integrated model, where modifications are made to a schedule design rather than starting from scratch. The model takes as an input a master flight list that includes mandatory and optional flights, and the average unconstrained itinerary demands for the flights in the master flight list. The authors show that significant benefits are probable with preliminary results.

Next, the aircraft routing and crew scheduling problems are solved simultaneously in *Cordeau et al.* (2001). An advantage of solving these two models simultaneously is that a lower cost solution may be found since more options are available for the crew scheduling problem, which is normally solved after the aircraft routing problem, using its solution as input. Crews have a minimum connection time between any two flights, which depends on whether or not the same aircraft is flown for both flights. These connection times can be taken into consideration while planning aircraft routes. The difficulty though is that both problems have large data sets and can be time-consuming to solve one at a time, so solving them together presents an even bigger challenge. In the paper *Cohn and Barnhart* (2003), the authors suggest delaying some decisions in maintenance routing in order to keep more possibilities open for the crew scheduling decisions. They provide two solution methods, heuristics and finding the optimal solution.

In the paper *Mercier et al.* (2005), the authors solve the same two problems, and they handle the linking constraints which enforce rules on connection times for crews by using Benders decomposition. They compare using the aircraft routing problem as the master problem versus the crew scheduling problem as the master problem. In *Mercier and Soumis* (2007), the authors let flights be re-timed in a certain time window while solving the two problems simultaneously. Benders decomposition is used to solve the model in order to solve it more quickly or make it solvable. In all these examples, the authors find that their results are valuable and can be found in a reasonable amount of time.

In *Stojković and Soumis* (2001), the operational flight and pilot scheduling problems are solved simultaneously. The authors allow changes in the departure times of flights, while keeping itineraries unaffected, and plan the work days of pilots for one day of operations. The model is an integer nonlinear multi-commodity flow problem with extra constraints. The commodities are the pilots, and the extra constraints include enforcing integrality for the flow variables and handling the time windows for the flights.

The solutions from the fleet assignment problem, crew pairing and rostering problems, and other planning problems affect passengers greatly. These solutions determine what itineraries are possibilities for passengers, how many seats are available on each flight, and the price of each ticket. In addition, connection times between flights are determined, affecting the chances of crews and aircraft being late for flights and the ability of passengers to easily make connections. In particular, the maintenance routing problem affects passengers, since using two different aircraft for two flights determines possible delays differently than using the same aircraft for the two flights does. Even deciding what gates to have flights arrive at and depart from affects delays. Note that the models used for all of these problems cannot be easily adapted to solve the passenger recovery problem, since these models have different types of objectives, constraints, and decisions that need to be made, since they concern different factors and do not involve recovery. The differences for handling different factors are discussed more fully in section 2.2.

After the planning process is complete, airlines often face unexpected problems with their flights, so other optimization problems are solved after flight delays and cancellations have occurred in order to get an airline back on schedule. The issues of recovery and creating more robust schedules less prone to delays are studied more fully in chapter II, since they are more closely related to our research.

A thorough overview of airline planning is in the papers *Barnhart et al.* (2003) and *Barnhart and Cohn* (2004). Also, the paper *Clarke and Smith* (2004) describes in detail many ways operations research has been used successfully in the airline industry, explaining research activity, the benefits to the industry, and areas of research opportunity. A good discussion on the airline industry's current challenges is in *Garrow* (2009). Last, the book *Belobaba et al.* (2009) has a thorough description of many areas of the airline industry and progress made in each area, including regulations, markets, demand, pricing, measures of productivity, the airline planning process, schedule optimization, schedule recovery, and robustness.

1.3 Overview of Our Work

As mentioned, an important recovery problem for airlines when unexpected delays occur is handling the reaccommodation of passengers. This problem is difficult with many factors to take into consideration, such as the unknown length of delay on the flight being considered and on other flights, low available capacities on flights, communicating results to passengers, and prioritization of business-class passengers or frequent flyers. Disruptions can be caused by many different situations, including inclement weather, maintenance problems, late crews, or overcrowded airports with no available gates. These delays not only delay passengers on the current flight, but may also lead to missed connections. This can cause significant problems if the connections occur late in the day with little time to find alternatives or if the connections are international flights that occur infrequently. If flights are nearly full, reaccommodation is especially difficult. Even when a passenger has a direct flight that is delayed, he or she may miss a meeting or other event that was the purpose of the trip. Passengers may become disgruntled, and airlines may have to pay fines in extreme cases if they cannot reaccommodate passengers well.

Typically when a flight is delayed, airlines do not move passengers off the delayed flight unless it is canceled. After the flight is completed, any passengers that have missed connections are reaccommodated one-by-one. The order may be determined by preference based on frequent-flyer status or other factors, or may be determined by the order that passengers arrive at the desk. This method of handling passengers' experiences has limitations and could be improved upon in a few ways.

In our models, passengers are reaccommodated proactively as soon as it is known that a flight will be delayed, instead of waiting until the passengers have taken the flight and some have missed their connections. We assume that there are a finite number of possible lengths of delay and that the probability of each possibility is known. If there is a low probability of a delay, it may be better for passengers to wait for their original, delayed flight. If there is a high probability of a long delay, it makes sense to move them to new flights when possible.

In addition, we look at all of the affected passengers on one or more delayed flights together so that we can give certain seats to people who need them most. That way we can handle the issue that the passengers are all competing for the available seats. Those with priority can include people whose connections are soon or are international flights. Our models allow passengers who paid higher fares, are in first class seating, or are frequent flyers to be given more preference by multiplying their delay cost in the objective function by a constant.

We show through computational experiments in section 4.3.1 that our model from section 3.2.6 is quick to solve so passengers will be reaccommodated efficiently. Reasons for this are explained more fully in section 4.3. Ultimately, passengers will spend less time waiting for their flights to take off or get new itineraries, will miss fewer connections, and will arrive at their destinations sooner.

1.4 Contribution and Organization of Thesis

We now describe the organization of the thesis and contributions from the chapters. First, in chapter II, background information for our research is given. There is a literature review discussing work in creating schedules that are more robust which behave better under delay situations. In addition, we present research in recovering after delays, including handling aircraft, flight schedules, maintenance, crew, and passengers.

The contribution of chapter III includes several models that represent reaccommodation of airline passengers after delays, some of which take into account probabilities for the length of the flight delay in order to handle stochasticity. Specifically, in section 3.1, we give three models to reroute passengers after a flight delay has been discovered when the length of delay is known, along with explanations about the advantages and disadvantages of each model. In section 3.2, the length of delay for the flight is not known for certain. In this section, we give three methods to handle this stochasticity and present four two-stage stochastic programming models with the goal of rerouting passengers. Since the length of a delay is often not known in advance, we consider *preemptive rerouting* of airline passengers before the length of the delay is known. Our goal is to reaccommodate passengers proactively as soon as it is known that a flight will be delayed instead of waiting until passengers have missed connections and to use known probabilities for the length of delay. The model in section 3.2.6 in particular is useful because the second-stage problem is modeled as a minimum cost flow problem which always has integer solutions. We also present an analysis on the size of the different models, an algorithm for converting arc-based solutions from our models to path-based solutions that can be given to passengers, and some applications to other areas of research.

Some progress has been made in the area of recovery after operational disruptions for airlines. More research has been done with rescheduling flights, aircraft, and crew, but little has been done specifically with the passenger recovery problem. In particular, we did not find any papers where there is preemptive rerouting before the delay length is known or where there is stochasticity in the models, except for the papers *Yen and Birge* (2006) and *Rosenberger et al.* (2003), which are for crew and aircraft schedules, respectively. Using assumed probabilities for the length of delay based on the type of delay has been used rarely, and not ever for passengers in particular, as far as we could tell. Proactively handling passenger delays before misconnections have occurred and allowing passengers to get off a delayed flight is a new contribution. Making decisions now while considering possibilities for the future is an important new idea for the area of passenger recovery.

Our computational results given in chapter IV show the framework and model's effectiveness for reducing the length of passenger delays. We did not see these types of results in other papers. Significantly, we can solve the model in most reasonable situations in real time, allowing the method to be used effectively for airlines. Also in chapter IV, we present a description, derivation, and example of Benders decomposition, a powerful method for solving large linear programming problems. We use Benders decomposition to solve the combined model from section 3.2.6. In another section of chapter IV, we discuss the value of the results in terms of passengers' inconvenience.

The contribution of chapter V includes the *portfolio model* for the reaccommodation of passengers from two or more simultaneous delays, where passengers are assigned *portfolios* that define their itineraries under all possible disruption outcomes. We present a strategic method of solving the model to optimality using branch and price, which involves both delayed column generation and branch and bound, specifically adapted for this model.

Last, in chapter VI, we have a description of future research plans.

CHAPTER II

Literature Review

After the airline planning process is complete, airlines often face unexpected problems with their flights, caused by bad weather, late crews, and maintenance problems. Optimization problems are solved after flight delays and cancellations have occurred in order to get an airline back on schedule. In addition, it is often beneficial to focus on creating schedules that are less prone to propagated delays, rather than choosing schedules that perform best under deterministic departure times. We first discuss advances in creating more robust schedules, then look at recovery problems after delays.

2.1 Creating Robust Schedules

Deterministic scheduling models do not necessarily give solutions that perform best in practice. An important area of research is creating more robust schedules that respond better under delay situations. We discuss some important results achieved.

Significant research has been done on better understanding delays and how they affect future flights, an airline's profits, passengers, and other factors. When one flight is delayed, it often causes the delay of other flights, called propagated delays. Some causes of propagated delays include crews usually being scheduled for several flights in a row, gates being unavailable because of delays, or runways being congested. Also, if a particular aircraft is scheduled to fly two flights back-to-back and the first is delay, the second is automatically delayed.

The author of *Janić* (2005) models the economic consequences of disruptions that affect many flights, evaluating the cost from delays and cancellations. Affected flights in turn have effects on airlines, airports, and passengers, and the cost of disruptions are on the order of millions of dollars. Costs include revenue from canceled flights, moving aircraft and crew during irregular operations, and compensating affected passengers. The author uses the idea of queuing systems, where the complexes of flights at a hub airport are the customers and the hub is the server, in order to accurately capture the effects of propagated delays. He also evaluates the complexity of this type of problem and the effect of different factors.

The authors of *Hsiao and Hansen* (2006) formulate a model of the average daily flight delay, considering factors including queuing, especially time-of-day effects of arrivals, and weather at airports and on flights' routes. Through queuing theory, the authors find that a certain amount of queuing delay in the morning causes a much bigger impact on average daily delay than the same amount of delay in the evening does.

The authors of *Lapp et al.* (2008) formulate a way to study how robust a given flight schedule is in order to determine how delays can be propagated from one flight to others. The goal is to reduce this propagation so less flights are delayed by creating flight schedules and other factors in a more strategic way. Their recursion-based approach enables them to overcome challenges in simulating this complicated situation of propagated delays. The focus of the authors of *AhmadBeygi et al.* (2008) is on passengers and their potential flights. Through the authors' work, they better understand the scheduling of crews and aircraft and the operational performance of a schedule affected by propagated delays. One conclusion is that keeping a certain crew with a specific airplane helps to reduce delay propagation substantially, although the authors mention that they do not know how this could affect delay propagation since slack may be reduced between flights. Also, when the original delay occurs early in the day, it often causes more problems, but adding slack in the middle of the day may be more beneficially when considering all factors.

The paper *Lan et al.* (2006) is one of several that discuss redistributing slack between flights in an airline's schedule, where the changes are small enough not to affect other decisions already made. Redistributing slack does not necessarily contribute cost to an airline, but can significantly decrease the number of missed connections for passengers by decreasing the propagation of delays. The authors present two models with the goal of minimizing passenger disruptions. The first reroutes aircraft to reduce expected delay propagation by allocating the slack in the system to the places where it is most needed. The second model minimizes misconnections by giving a small window to each leg and choosing the best time for departure. In *Fuhr* (2007), the author uses a stochastic model for the problem of creating schedules that react better after disruptions and also promote productivity, and uses an analytic approach rather than Monte Carlo simulations to decide on block times and ground times for flights.

In the paper *AhmadBeygi et al.* (2010), the authors redistribute slack already existing in the system in order to lessen delay propagation. They allow small changes in the flight departure times, but they do not allow changes in the fleet assignment solution or the crew scheduling solution, so that planned costs do not change at all, but operational performance can improve. The slack can be arranged around flights that are most affected by disruptions and have a higher chance of causing delay propagation. Fortunately, the models can be solved ignoring integrality constraints. One powerful model they present is the multi-layer model, where propagated delays carried to many other levels are considered until delays are completed absorbed. In the single-layer model, only propagated delays carried one layer downstream, flights that connect from the original delayed flight, are considered.

In *Burke et al.* (2010), the authors simultaneously re-time flights and handle aircraft routing, where the flight assignment is not allowed to change. They use a multi-meme memetic algorithm in creating more robust schedules, and show significant improvements in the reliability and flexibility of the schedules.

In the paper *Ehrgott and Ryan* (2002), the authors discuss the conflict between a low planned cost for an airline and less delay propagation, and they create a bi-criteria optimization framework considering both of these goals, specifically for crew scheduling. In the model, crew scheduling solutions that are not robust are penalized. The penalization is determined by considering scheduled ground time minus ground duty time, such as breaks for the crew, compared to the expected delay. If it is less than the expected delay, the solution is penalized. They find that their method can produce much more robust schedules with a small increase in cost. The authors of the paper *Schaefer et al.* (2005) propose methods to find crew schedules that perform better in practice, where they consider "frictional" delays, which are of limited length, showing that their methodology performs well.

The work in the first four papers mentioned in this section helps us to understand better the effects of the timing and lengths of delays on flights, profits, passengers, and other factors and how these effects may be mitigated. The other papers discussing strategies to create most robust schedules, including those that redistribute slack in the system, can help create significantly better original schedules that reduce the length of delays and number of missed connections experienced by passengers. This work does not give a clear picture of how to recover from delays that unfortunately do occur even when schedules are designed well, including reaccommodation of aircraft, crews, and passengers.

2.2 Recovering After Delays

When a flight is delayed or canceled, the flow balance for the aircraft and crew must be preserved, and the passengers must be reaccommodated. Aircraft may be rerouted in case of cancellations or if delays are long enough that substitute aircraft need to be used, and it is important to keep each aircraft maintained at the required times. A feasible schedule is needed, taking into account different aircraft capacities and types. There are strict, complicated work rules for crews, governing the mix of work time and rest time. If a flight is delayed, these rules may be violated, so back-up crews may be called. When reaccommodating passengers, it is important to get them to their destinations as soon as possible, but unknown delays and limited capacities on other flights make this problem difficult. Not much research has been done specifically in the area of passenger reaccommodation after a delay. We discuss some advances in the area of recovery after delays.

Handling decisions about aircraft is an important area of research. It is discussed by the authors of *Jarrah et al.* (1993), particularly pertaining to aircraft shortages. They swap aircraft, bring in extra aircraft, and delay and cancel flights in real time, where the flight schedule is kept feasible. In *Yan and Yang* (1996), the authors also study how to help airlines recover from small changes in their flight schedules due to the breakdown of airplanes. Their models are network flow problems, sometimes with side constraints, used to reroute aircraft and reschedule flights as needed. In a similar way, in *Cao and Kanafani* (1997a) and *Cao and Kanafani* (1997b), there is a quadratic binary programming model and also a continuous model to help with decisions about the flight schedule and aircraft assignment, including bringing in surplus aircraft. Computational experiments are shown.

In *Jonathan et al.* (2001), the authors present a method for reconstructing aircraft routings after delays, where the cost involves delays and cancellations produced by decisions made. They use a formulation that is an integral minimum cost flow problem with side constraints, whose solutions must be transformed to give a solution for aircraft routings. In *Rosenberger et al.* (2002), the authors consider passenger inconvenience as one way to measure the quality of a schedule by counting the number of passenger misconnections from flights being canceled or delayed as they make decisions about which flights to cancel.

There are some similarities between aircraft recovery and passenger recovery, the topic of this dissertation, since some of the same information is used for both problems and resources are the same. Unfortunately, we cannot easily adapt the models for aircraft recovery to help with passenger recovery, especially considering preemptive rerouting under uncertain delays, for several reasons. First, the objectives are much different. The goal for passenger recovery is to have the passengers arrive at their destinations as soon as possible, and for aircraft recovery the goal usually involves minimizing the airline's cost of operating the aircraft chosen. The types of constraints are also different. Aircraft must have different types of maintenance at certain intervals, but passengers only need to reach their destinations while not exceeding flight capacities. Decisions about aircraft affect future flights since one aircraft is often used several times in one day. If an aircraft is used for a delayed flight, it may not be able to be used for other flights. Also, there is the option of canceling flights, which may affect much of the flight schedule. Decisions made for passengers have no propagated effects, except that two passengers cannot both be assigned the same seat. Last, the papers discussed do not look at proactive recovery, two-stage decisions, or delay probabilities.

Next, much research has been done on giving new schedules to a crew after a flight is delayed or canceled, which is called the crew recovery problem. The Crew Recovery Model is given in *Lettovsky* (1997) and *Lettovsky et al.* (2000) that solves this problem. There is a separate model for each fleet type or set of fleet types. One component of the cost function is the cost of rerouting passengers, feeding them, giving them hotel rooms, and the loss of their goodwill. The model is a set covering model, where the variables represent assigning groups of flights to crews and canceling flight legs, where each flight leg must be assigned a crew and crews can take flights without working on them, called deadheading. Since the variables are integral, the primal-dual subproblem simplex method is used to solve the LP relaxation.

In the paper *Abdelghany et al.* (2004), the authors present a tool for decision-making that proactively handles reaccommodation of crews. Their goal is to minimize cost from reassignments and delays. An example of a stochastic integer programming problem with recourse is given to solve the airline crew scheduling problem in the paper *Yen and Birge* (2006). In the objective function of the model is the cost if the problem were deterministic, and there is also a term for the expected cost of recourse in case of disruptions. There is a set of possible disruption scenarios Ω with finite size where each element $\omega \in \Omega$ has a known probability. The goal is to proactively take into account possible disruptions by finding original crew schedules that are more robust.

Crew recovery, like aircraft recovery, has some significant differences from passenger recovery. Specifically, there are complicated work-time rules that must be followed for crew, such as how many hours of work are allowed in a certain time period. There are not normally counterparts to these rules for passengers. Also, similar to aircraft recovery, decisions made about crews may affect many flights, since a crew often handles several flights in a row. Decisions even include calling in back-up crews, but for passengers, these considerations are not relevant. The goals for crew recovery and passenger recovery are different as well. Interestingly, the two papers *Abdelghany et al.* (2004) and *Yen and Birge* (2006) discuss preemptive rerouting of crews, which is similar to our work concerning passengers in some ways, but the models cannot easily be used for passengers because of the differences described.

Research has also been done where authors suggest ways to recover multiple factors. In the work *Clarke et al.* (1998), the authors consider rescheduling planned flights after delays for transportation systems in general, including airlines. The authors give a formulation for the airline schedule recovery problem and give heuristics for yield management, vehicle routing, maintenance scheduling, and crew scheduling. The three steps in their heuristics are generating flight sequences with modified tree search algorithms, assigning sequences of flights to aircraft to optimize profit or other goals, and revising arrival and departure times of flights. In the paper *Rosenberger et al.* (2003), the authors have a model where the decisions are what time the flights should be and how to reroute aircraft, and the goal is to minimize a function of rerouting and cancellation costs. They also have a model that minimizes crew and passenger disruptions. In addition, in their section about future research, they discuss using a two-stage stochastic programming problem, taking into account the weather possibilities, to solve the aircraft recovery problem.

In the two models in the paper *Bratu and Barnhart* (2006), the goal is to balance airline costs and passengers' delay costs. Several decisions are made in order to choose flight departure times and cancellations and reschedule aircraft, crew, and passengers after a delay or disruption has occurred. The first model minimizes operating costs and passenger disruption costs and the second minimizes operating costs and passenger delay costs. One of the models can be solved in real-time, and the models give noticeable reductions in passenger delays and disruptions. For example, with a case study where there was a higher level of disruption than average, the number of passengers disrupted overnight decreased by about 63% and the delay experienced by passengers decreased by about 16%.

In the paper *Eggenberg et al.* (2010), the authors give a modeling framework, presenting a "recovery network," which can be used for different factors to be reaccommodated. Each feasible recovery method corresponds to a path through the nodes and arcs in the network, which has a two-dimensional coordinate system (A, t). A is horizontal and represents locations of airports, and t is vertical and represents time and is discretized. A node corresponds to a value $a \in A$ and a time t, and there is one source and several sink nodes representing the initial state of the system and eligible final states. The authors specifically show solving the aircraft recovery problem and also discuss handling the passenger recovery problem using their approach.

In the paper *Jafari and Hessameddin Zegordi* (2010), a model is solved after a disruption to reroute both aircraft and passengers simultaneously. The decisions include re-timing flights, switching airplanes, and giving passengers new itineraries. The authors use aircraft rotations and itineraries instead of flights for passengers and include an option of reaccommodating passengers with other airlines or other transportation methods.

In the paper *Petersen et al.* (2010), the authors solve simultaneously all the recovery problems for the flight schedule, aircraft routings, crew schedules, and passengers' itineraries. This is advantageous, because a solution for a previous step may not produce the optimal solution for a subsequent step. Since solving this large MIP problem takes too long in most cases, the authors use Benders decomposition, delayed column generation, and heuristics to decide which flights can be ignored that are not affected by the delays, and other methods. Delayed column generation is a method used when there are many variables. Some columns (variables) are ignored, the problem is solved, then all unused columns are checked to see if using them could improve the solution at all, and the problem is re-solved. Many iterations may be completed to find the optimal solution. In this paper, the passenger recovery problem is modeled as a multi-commodity flow problem where flights correspond to arcs and passengers are assumed to be homogeneous. The objective function is the sum of the amount of time delayed over all passengers plus the cost of not reaccommodating passengers.

The authors are able to solve an example with a major carrier and find the solution time is realistic.

The paper *Clausen et al.* (2001) discusses disruption management in a few different fields. The Descartes project has the goal of creating a tool to help an airline with decision-making after disruptions have occurred for rerouting aircraft, flight crew, cabin crew, and passengers. After a delay, a consequence analyzer simulates the results from different decisions and strategies for the rest of the day, and a stochastic simulator analyzes potential overall strategies to deal with disruptions. Also, alerting mechanisms show when a significant problem may occur since disruptions may result from several small events, such as many crew members calling in sick for the day, rather than just one bigger event

These papers on recovering multiple factors all discuss passenger reaccommodation after delays at some level. Still, none of them discusses preemptive reaccommodation, considers two-stage decisions, or uses probabilities to make better decisions, so cannot be used for our problem. The only exception is that in *Rosenberger et al.* (2003), the authors briefly suggest a method for preemptive rerouting for aircraft recovery in the future research section, which I did not find in a later paper. The idea is interesting and similar to principle to our work. Since it is for aircraft rather than passengers, any model and method would have significant differences.

The authors of *Luo and Yu* (1997) discuss the ground delay program by the Federal Aviation Administration and how it affects airline schedule changes. As mentioned in the abstract, this program was used "for efficient and equitable use of scarce airspace and airport capacity." For example, when there is inclement weather, the FAA might decide that there are too many planned arrivals at a particular airport, and flights must be delayed. Arrival times of flights are mandated, which may cause serious problems for crew schedules and aircraft maintenance schedules. When this occurs, the authors' goal is to reduce the number of flights more than fifteen minutes late.

They use an integer model, which they show significantly decreases these delays. The author of *Carlson* (2000) discusses collaborative decision making (CDM) which replaced the ground delay program and helps with traffic flow management in certain conditions. He gives four formulations of an integer model to help airlines with decisions about rescheduling at their hub airports.

For a more comprehensive review of recovery models for flight schedules, aircraft, crew, and passengers, see *Bratu and Barnhart* (2006), *Kohl et al.* (2007), and *Petersen et al.* (2010). The book *Yu and Qi* (2004) is a good resource for better understanding disruptions in the airline industry and recovery. Chapter nine of the book *Belobaba et al.* (2009) titled "Irregular Operations: Schedule Recovery and Robustness" discuss many of these topics in depth.

CHAPTER III

One Flight Delayed

3.1 Deterministic Problem

To understand passenger reaccommodation better, we first consider a deterministic problem, where one flight has been delayed by a known amount of time. We create a list of possible alternative itineraries for each passenger, each of which has a deterministic and known arrival time. The goal is to minimize a function of the cost or dissatisfaction over all passengers on the original flight. The decisions at the time of delay are which passengers should stay with their current itineraries, which passengers should get new itineraries, and what those should be. A passenger may keep the current itinerary only if he or she has no chance of missing a connection with that itinerary.

We make a rule that all passengers must arrive at their destinations by a certain end time, so that we can restrict the size of our data set to decrease the solve time and to promote equality between passengers. If the problem is infeasible with the chosen end time, we can increase the end time and re-solve the problem. In the unlikely case that there is no way to make the problem feasible, then the airline would have to use another method, such as putting some passengers on another airline's flights. Also, each passenger's itinerary can have no more than three flights. The reason for this is to decrease the complexity of the problem and to promote equality between passengers as well. After the decisions are made, each passenger is then told what his or her itinerary will be.

The cost of an itinerary could be the difference between a passenger's originally scheduled arrival time at the desired destination and the arrival time for the newly-assigned alternate. Other choices for how to measure the cost are described in section 3.1.1. More weight can be given to certain passengers, such as those who fly frequently or who paid more for their tickets, by multiplying their delay cost by a constant greater than 1. Note that the way cost is measured for a passenger is a function of only the arrival time in the final destination and possibly a constant factor. No other things, such as flight time or layover location, are considered.

We have formulated three different models for this deterministic problem. The first is an itinerary-based model, the second is a flight-based model that is a multi-commodity flow formulation, and the third is another flight-based model that is a minimum cost flow formulation. Each of these has different advantages and disadvantages.

First, the itinerary-based model has binary variables that represent a passenger being assigned a particular itinerary. An advantage of this model is that it is easier to understand and is more concise than the others. In addition, it has fewer constraints than the other models do, as the experiments in section 3.1.4 illustrate. This model can be expanded well into a two-stage problem where the length of delay is not known, as explained in section 3.2. Also, we need the itinerary-based formulation for part of the combined model in section 3.2, in order for the objective function to be linear. On the other hand, a disadvantage of the itinerary-based model is that all variables are binary and the integrality constraints must be enforced when solving the model. In addition, for larger-sized models, it has more variables than the minimum cost flow model does, as shown in section 3.1.4.

Second, in the first flight-based model, we create a network to represent passengers taking flights. Each variable represents flow on a certain arc for a particular passenger. An advantage of this formulation is that it can be used to model well the two-stage problem in section 3.2. Also, unlike with the minimum cost flow model, the solution clearly shows what flights to give to each passenger, and we do not have to solve a separate flow-to-path problem. A disadvantage of this model is that integrality constraints must be enforced.

Third, the network for the miminum cost flow formulation is similar to that for the multi-commodity flow formulation, except all passengers are treated as one commodity. The variables represent total flow over the arcs. This model is useful because solutions are automatically integral without enforcing integrality constraints since the supplies and capacities are all integer. The proof of this fact is in the document *Van Roy and Mason* (2005) on page 109. Also, this formulation almost always leads to a model with less variables and constraints, since there is not a set of variables and constraints for every passenger, and it is easier and faster to solve than the others. Last, we need to use the minimum cost flow formulation for our preferred two-stage model in section 3.2. A disadvantage is that we have to solve a separate flow-to-path problem after solving for the variables.

A list and description of all of the sets, parameters, and variables for all three models are given in appendix A.

3.1.1 Itinerary-based Model

In the itinerary-based formulation, the possible itineraries for all the passengers are found ahead of time by looking at all the possible flights and enforcing several rules on each itinerary. The model is described after these rules.

- There may be one, two, or three flights.

- Each passenger must arrive at the location he or she desires.

- The last flight must arrive by a pre-set end time.
- The first flight must start in the original location of the delayed flight.
- The first flight must take off no earlier than the originally scheduled departure time of the delayed flight.
- A second or third flight must begin at the previous flight's destination.
- A second or third flight must take off no earlier than the time the previous flight arrives plus a certain amount of time for de-boarding and boarding that depends on the airport.
- The second and third flights cannot arrive at the passenger's original location.

Let P be the set of passengers on the delayed flight.
Let F be the set of all flights in the data set.
Let I_p be the set of all possible itineraries for passenger $p \in P$.
Let $F_i \in F$ be the set of all flights from itinerary $i \in I_p$ for $p \in P$.
Let C_f be the available capacity of the aircraft assigned to flight $f \in F$.
Let c_{pi} be the cost of delay for passenger $p \in P$ if reassigned to itinerary $i \in I_p$.
Let x_{pi} be a decision variable that is 1 if passenger $p \in P$ is assigned to itinerary $i \in I_p$ and is 0 otherwise.

There are three sets of constraints that must be satisfied, given below. The first set ensures that every passenger is assigned to exactly one itinerary. Any itinerary that satisfies the requirements above and includes the original flight, such as the passenger's original itinerary, is an option for the passenger only if the itinerary will not be disrupted with the delay. The next set of constraints says that we cannot assign more passengers to a flight than there are available seats, and the last makes all decision variables be binary.

$$\sum_{i \in I_p} x_{pi} = 1 \text{ for all } p \in P \tag{3.1}$$

$$\sum_{p \in P} \sum_{i \in I_p : f \in F_i} x_{pi} \leq C_f \text{ for all } f \in F \tag{3.2}$$

$$x_{pi} \in \{0, 1\} \text{ for all } i \in I_p,\ p \in P \tag{3.3}$$

Next there are two possible objective functions we consider, each of which is minimized.

- $\sum_{p \in P} \sum_{i \in I_p} c_{pi} x_{pi}$
  Using this objective function, the goal is to minimize the sum of the delay over all passengers. Different weights can be given to different passengers in the constants c_{pi}.

- $\max_{p \in P} \left(\sum_{i \in I_p} c_{pi} x_{pi} \right)$
  Using this objective function, the goal is to minimize the worst cost incurred by any one passenger. This objective function promotes equity among passengers. It is nonlinear, but we can name the objective z and add the constraints $z \geq \sum_{i \in I_p} c_{pi} x_{pi}$ for all $p \in P$ so that the problem is linear.

We prefer to use the first objective function, so that we can lower the delay for everyone as much as possible, but having more equality between passengers is valuable. Thus, we can use the first objective function, but add one of these two constraints to the model.

- $\max_{p \in P}\left(\sum_{i \in I_p} c_{pi} x_{pi}\right) \leq cap.$

  This makes the worst cost for any passenger not more than a pre-defined amount *cap*. Since this constraint is not linear, we substitute it with the following constraints.

  $z \leq cap$

  $z \geq \sum_{i \in I_p} c_{pi} x_{pi}, \quad \forall p \in P$

- $\max_{p \in P}\left(\sum_{i \in I_p} c_{pi} x_{pi}\right) - \min_{p \in P}\left(\sum_{i \in I_p} c_{pi} x_{pi}\right) \leq dif.$

  This makes the difference between the highest and lowest cost for any two passengers be bounded by a pre-defined amount *dif*. Since this constraint is not linear, we substitute it with the following constraints.

  $z_1 - z_2 \leq dif$

  $z_1 \geq \sum_{i \in I_p} c_{pi} x_{pi}, \quad \forall p \in P$

  $z_2 \leq \sum_{i \in I_p} c_{pi} x_{pi}, \quad \forall p \in P$

3.1.2 Flight-based Model

We have also modeled the problem as a multi-commodity flow problem, where each passenger is a separate commodity. In our model, there are three types of nodes.

- A start node, representing the original location, that has a supply of 1 for each passenger.

- A pair of flight nodes for each flight in the data set so that we can set a capacity on the arc between the two nodes. Each of these nodes has a supply of 0.

- One end node that represents arrival and has a demand of 1 for each passenger.

There are also several types of arcs.

- For each flight, an arc from the start node to the first of the pair of flight nodes under certain conditions. First, the departure city of the flight must be the same as the current location. Second, the flight has to depart after the scheduled departure time of the original flight. Last, the flight must arrive before the specified end time before which all passengers must arrive at their destinations.

- Arcs between two flight nodes in a pair with capacities equal to the available capacity of the relevant flight. We call this set of arcs A_{fp}.

- Arcs between the second of one pair of flight nodes and the first of another pair of flight nodes if it is possible to connect from the first flight to the second flight. This is true if the second flight departs from the arrival location of the first flight, takes off at least a certain amount of time after the first arrives (which depends on the airport), and arrives before the end time. In addition, the second flight should not arrive at the origin of the delayed flight since taking that flight would then never be beneficial to passengers.

- Arcs A_e going to the end node. For a pair of flight nodes, the second node in the pair connects to the end node if the arrival location for the flight is the same as some passenger's desired destination and the flight arrives before the pre-set end time. For a passenger $p \in P$ and $a \in A_e$, if the arrival location of the corresponding flight is not the same as the passenger's desired destination, then the capacity for that passenger on the arc is 0; otherwise, the capacity for that passenger is infinite. Thus we have passenger-specific capacities u_a^p for any arc $a \in A_e$ for all $p \in P$.

The only cost for the passengers is the cost of delay, measured as the difference between their actual arrival times and their planned arrival times, where this cost cannot be negative. We thus assign a cost to using the arcs in A_e, which represent landing in the desired location after taking the last flight in a passenger's itinerary. Note that this cost is specific to each passenger, since passengers have different planned arrival times. For example, consider a passenger p that planned to arrive at FLL at 2:30pm. If there is an arc a from the second of a pair of flight nodes for a flight that lands at FLL at 3:30pm to the end node, then the cost c_a^p is 60. Arcs other than those in A_e have a cost of 0 since the only cost we consider is the final delay cost.

Unless otherwise stated, the capacity of an arc is infinite. We only need to control the number of passengers placed on a flight based on the number of seats available and make sure that a passenger goes to the end node only when he or she lands at the desired location. For example, on the arcs between two flight node pairs, there is no reason to restrict the number of passengers that can use those arcs since they are already limited by the capacities on the flights.

A picture showing the network for some of the flights in our data set described in section 4.2.1 is below in figure 3.1. Note that above each arc between two corresponding flight nodes is "u" with a subscript of the flight number. This is the capacity of the flight. Along with the picture is a description of the model.

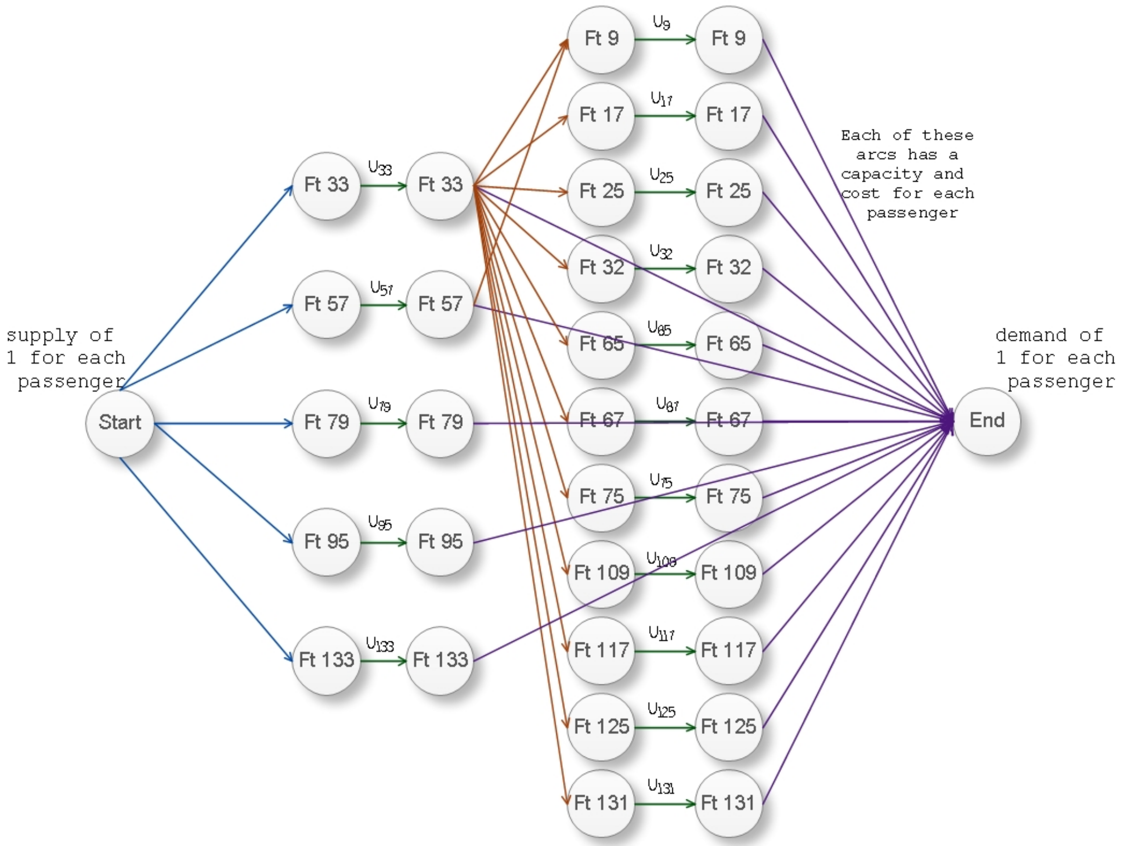

Figure 3.1: Multi-Commodity Flow Network

Let P be the set of passengers (commodities).

Let N be the set of all nodes.

Let A be the set of all arcs.

Let $A_e \subset A$ be the set of arcs going to the end node.

Let $A_{fp} \subset A$ be the set of arcs between two flight nodes in a pair.

Let B_n be the set of arcs that begin at node $n \in N$.

Let E_n be the set of arcs that end at node $n \in N$.

Let c_a^p be the cost for passenger $p \in P$ on arc $a \in A_e$.

Let s_n^p be the supply for passenger $p \in P$ at node $n \in N$.

Let u_a^p be the capacity of arc $a \in A_e$ for passenger $p \in P$.

Let u_a be the total capacity of arc $a \in A_{fp}$.

Let x_a^p be 1 if arc $a \in A$ is used for passenger $p \in P$ and 0 otherwise.

The goal of the problem is to minimize the delay cost for the passengers. The objective function is $\sum_{p\in P}\sum_{a\in A_e} c_a^p x_a^p$. The cost is summed over only the arcs in A_e since those are the only arcs with non-zero cost. The first sets of constraints, the balance constraints, is

$$\sum_{a\in B_n} x_a^p - \sum_{a\in E_n} x_a^p = s_n^p \text{ for all } n \in N,\, p \in P \tag{3.4}$$

At each node, the supply plus what comes in equals the demand plus what goes out. Note that the right hand side is 1 for the start node, $-|P|$ at the end node, and 0 at all other nodes. The second and third set of constraints are

$$x_a^p \leq u_a^p \text{ for all } a \in A_e,\, p \in P \tag{3.5}$$

$$\sum_{p\in P} x_a^p \leq u_a \text{ for all } a \in A_{fp} \tag{3.6}$$

These ensure that the capacity of each arc is not exceeded. Specifically, the second set puts capacities on the arcs that end at the end node, which ensure that passengers arrive at their desired destinations. For example, say that passenger p wants to arrive at DTW. If flight 1234 lands anywhere but at DTW, then the capacity u_a^p for passenger p on the arc a from the flight's second node to the end node is 0. Last, the decision variables are all binary, so

$$x_a^p \in \{0, 1\} \text{ for all } a \in A,\, p \in P \tag{3.7}$$

Any path that satisfies the constraints makes up a viable itinerary for a passenger.

As in the itinerary-based formulation in section 3.1.1, we could use a different objective function than the one in the model above. In order to minimize the greatest cost for any passenger, we could use $\max_{p \in P}\left(\sum_{a \in A_e} c_a^p x_a^p\right)$ as the objective function. Alternatively, while using the objective function $\sum_{p \in P}\sum_{a \in A_e} c_a^p x_a^p$, we could add one of these two constraints to the model.

- $\max_{p \in P}\left(\sum_{a \in A_e} c_a^p x_a^p\right) \leq cap$, where cap is a pre-defined constant, to put a limit on the greatest cost for a passenger. Since this constraint is not linear, we substitute it with the following constraints.

  $z \leq cap$

  $z \geq \sum_{a \in A_e} c_a^p x_a^p, \quad \forall p \in P$

- $\max_{p \in P}\left(\sum_{a \in A_e} c_a^p x_a^p\right) - \min_{p \in P}\left(\sum_{a \in A_e} c_a^p x_a^p\right) \leq dif$, where dif is another pre-defined constant, in order to lower the difference between the highest and lowest cost different passengers experience. Since this constraint is not linear, we substitute it with the following constraints.

  $z_1 - z_2 \leq dif$

  $z_1 \geq \sum_{a \in A_e} c_a^p x_a^p, \quad \forall p \in P$

  $z_2 \leq \sum_{a \in A_e} c_a^p x_a^p, \quad \forall p \in P$

3.1.3 Minimum Cost Flow Model

The problem can also be modeled as a minimum cost flow problem, where all the passengers are one commodity. The minimum cost flow formulation is very similar to the multi-commodity flow formulation in section 3.1.2. The difference is that all passengers are treated as one commodity, instead of each passenger being his or her own commodity. Because of this distinction, there are several changes to the network and list of sets, parameters, and variables.

- In order to differentiate between passengers so we can get them all to their particular destinations and be able to give certain passengers priority, we have an end node for each passenger which represents arrival, instead of having just one end node. Each of these end nodes has a demand of 1.

- A_e is now the set of arcs going to any end node.

- For arc $a \in A_e$, we use the parameter c_a instead of c_a^p, since we do not have multiple commodities, to represent the cost of delay.

- For arc $a \in A_e$, we no longer have capacities. To make sure everyone gets to the desired destination, we instead have the rule that the only arcs that go to the end node for passenger p come from flight nodes that land where passenger p wants to arrive.

- We let s_n, instead of s_n^p, be the supply at node $n \in N$.

- We let x_a be the number of units of flow (passengers) sent across arc $a \in A$, instead of using x_a^p.

A picture showing the network for some of the flights in our data set described in section 4.2.1 is figure 3.2. Note that above each arc between two corresponding flight nodes is “u” with a subscript of the flight number. This is the capacity of the flight. Along with the picture is a description of the model.

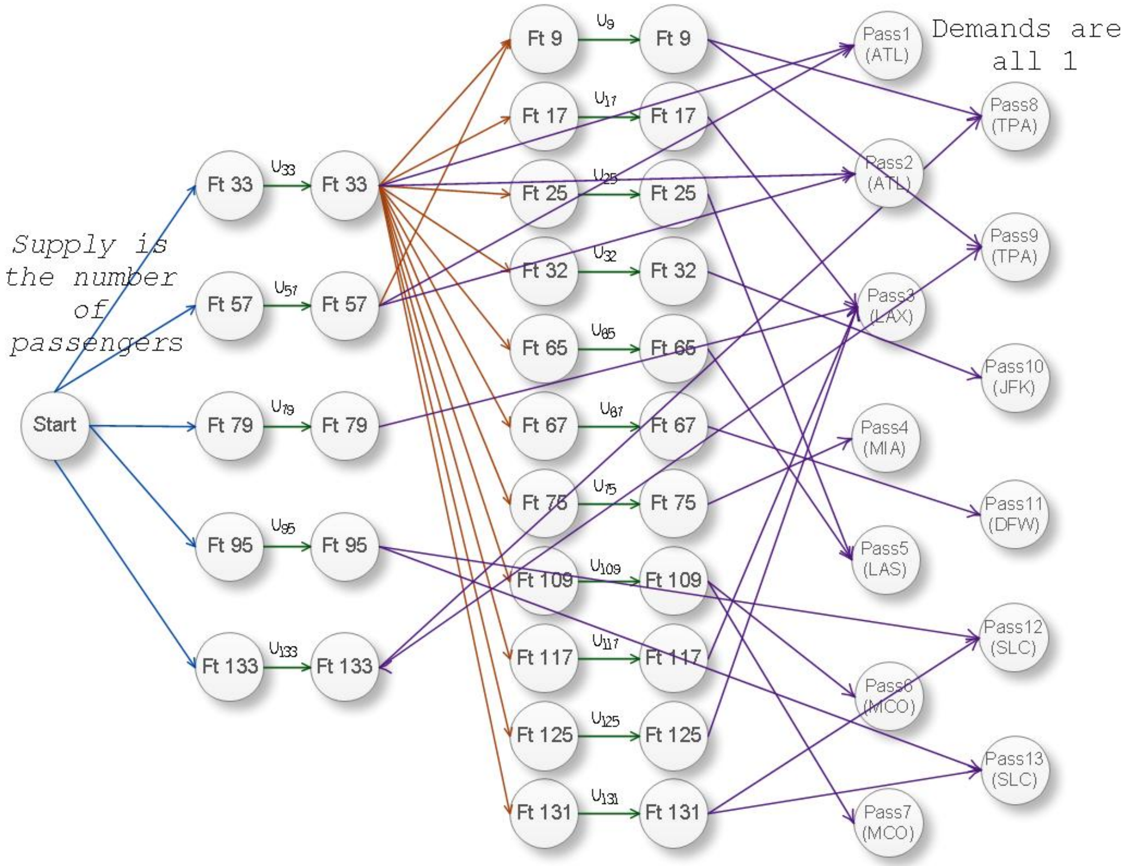

Figure 3.2: Minimum Cost Flow Network

The goal is minimizing the delay cost for all passengers. The objective function is

$$\sum_{a \in A_e} c_a x_a \tag{3.8}$$

The cost is summed over only the arcs in A_e since those are the only arcs with non-zero cost. The first set of constraints

$$\sum_{a \in B_n} x_a - \sum_{a \in E_n} x_a = s_n, \quad \forall n \in N \tag{3.9}$$

ensure that all passengers leave the original location and reach their desired destinations. At each node, the supply plus what comes in equals the demand plus what goes out. Note that the right hand side is 1 for the start node, −1 for each end node, and 0 at all other nodes. The second set of constraints ensures that no more passengers are put on a flight than there are available seats. These constraints are

$$x_a \leq u_a, \quad \forall a \in A_{fp} \tag{3.10}$$

Last, there cannot be a negative flow across any arc, since a person cannot take a flight in reverse, so

$$x_a \geq 0, \quad \forall a \in A \tag{3.11}$$

Recall that we can omit the integrality constraints on the variables, since all basic feasible solutions are integral because we have a minimum cost flow model, as explained in the introduction to section 3.1. A solution to this model provides us with the amount of flow on every arc. How can we determine what flights to assign to each passenger from this information? We can trace the flow backward from the end nodes. For example, we look at the end node corresponding to the first passenger. Starting at that node, we follow any positive flow backwards to another node. We keep following positive flow until we reach the start node. This gives the flights for the first passenger. Subtract the flow used for that passenger, then follow the same procedure to find the flights for the other passengers. In section 3.2.7, we describe this method in detail and prove that the path-based solution found from a given arc-based solution is unique. Also, we give an algorithm to find the path-based solution.

3.1.4 Size of the Three Models

So far, we have seen three models that accurately represent our deterministic problem. Next, we compare the size of the problem in the three different formulations, including the number of variables and number of constraints. First, we have a theoretical discussion and then some simulations.

3.1.4.1 Theoretical Analysis

First, for the itinerary-based formulation in section 3.1.1, there is a variable for each possible itinerary for each passenger. Letting F and P be the number of flights and passengers, respectively, the number of variables is order $O\big(F(F-1)(F-2)P\big) = O(F^3P)$, since itineraries can be composed of up to 3 flights. This has potential to be a rather large number depending on the scheduling of flights.

Next, consider the constraints. In the first set of constraints (3.1) in the itinerary-based formulation, there is one constraint for each passenger, which is a small number, P. The number of constraints in the second set (3.2) is the number of flights that are in at least one passenger's valid itinerary, which is $O(F)$. This is likely much less than the number of flights in the data set, since many flights are too early in the day or are not in the right location for any passengers on the delayed flight to take. Thus the number of constraints is of order $O(F + P)$.

Second, for the flight-based model in section 3.1.2, for each arc there is a variable for each passenger that may use that arc in some valid path from the start node to the passenger's end node. This is order $O\Big(\big(F+F(F-1)+F(F-1)(F-2)+F+F\big)\Big)P = O(F^3P)$.

The number of constraints in the first set (3.4) for one passenger is twice the number of flights in some valid itinerary for that passenger plus 2, counting the passenger's end node and the start node, so the total number in that set is order $O\big((2F + 2)P\big) = O(FP)$. In section 3.1.2, we defined a variable for each passenger for all arcs in the network. Since in this section we are considering only those arcs that are part of valid paths, we do not need the second set of constraints. In the third set (3.6), there is one constraint for each arc that some passenger may use that goes between two corresponding flight nodes, which is $O(F)$. Thus the total number of constraints is of order $O(FP)$.

Third, in the minimum cost flow model in section 3.1.3, there is a variable for every arc that is in a valid path for at least one passenger from the start node to the person's end node. This number is order $O\big(F + F(F-1) + F(F-1)(F-2) + F + FP\big) = O(F^3 + FP)$.

The number of balance constraints is twice the number of flights that are part of a valid path for at least one person, plus 1 for the start node, plus the number of passengers. This number is order $O(2F + P + 1) = O(F + P)$. The number of constraints in the second set is the number of flights that are part of at least one passenger's valid path, so it is order $O(F)$. Thus the total number of constraints is order $O(F + P)$.

Note that in the itinerary-based model and flight-based model, the variables are binary, so there are constraints that all variables are less than or equal to 1. In the itinerary-based model the first set of constraints actually enforces this. In the flight-based model, the balance constraints enforce this, so we do not need to consider these constraints. These finding are summarized in table 3.1.

Table 3.1: Analysis on the Size of the Models

Data/Model	Itinerary-based	Multi-commodity flow	Minimum cost flow
# of Variables	$O(F^3P)$	$O(F^3P)$	$O(F^3 + FP)$
# of Constraints	$O(F + P)$	$O(FP)$	$O(F + P)$

3.1.4.2 Simulations

We now discuss the results of five simulations, each with a different number of flights or passengers, in order to compare the number of variables and constraints in each of the three models. Recall that only flights that may be used by some passenger contribute to the number of variables and constraints. The simulations are described and then summarized in the table.

For our first four simulations, we use the data set described in section 4.2.1. In the first three, we use the whole set of 1144 flights, and in the fourth, we used only the first 572 flights. The delayed flight is flight 33, leaving at 8:20 am and traveling from JFK to ATL. Also, in the first, second, and fourth simulations, there are 100 passengers, and in the fourth, there are 50 passengers. In the first, second, and third, they are going to 15 different possible locations, and in the fourth, they are going to 30 different locations. In all these cases, about 1/3 of the passengers are going to ATL, and for each of the other locations, the passengers are evenly distributed. The delay is 90 minutes.

In the fifth simulation, we use a different data set with only 8 flights, also described in section 4.2.1. The delayed flight is flight 33, going from IAH to ATL. The three passengers are traveling to ATL, TPA, and PWM, and the delay is 90 minutes.

Tables 3.2 and 3.3 illustrate the differences in the data in the simulations and also gives the number of variables and constraints in each for each of the models, the itinerary-based model, the multi-commodity flow model, and the minimum cost flow model. Note that the row "Num Poss Fts" tells us how many flight of the total data set for the corresponding simulation are actually in a possible itinerary for at least one passenger.

Table 3.2: Simulations 1 to 3 for the Size of the Models

Data Info	Simulation 1			Simulation 2			Simulation 3		
Num Pass	100			50			100		
Num Fts	1144			1144			572		
Num Poss Fts	91			91			34		
Num Final Dests	15			15			15		
Delay	90			90			90		
Model	Itin	Multi	Mcf	Itin	Multi	Mcf	Itin	Multi	Mcf
Num Vars	2237	6769	1416	1108	3356	914	571	2219	454
Num Cons	191	6998	374	141	3472	324	134	2672	203

Table 3.3: Simulations 4 and 5 for the Size of the Models

Data Info	Simulation 4			Simulation 5		
Num Pass	100			3		
Num Fts	1144			8		
Num Poss Fts	154			6		
Num Final Dests	30			3		
Delay	90			90		
Model	Itin	Multi	Mcf	Itin	Multi	Mcf
Num Vars	2157	6703	1725	10	32	21
Num Cons	254	7022	563	9	39	22

The finding in section 3.1.4.1 are confirmed by the simulations. We highlight a couple of important points first.

- The multi-commodity flow model has many more variables and constraints than either of the other two models in all simulations. The number of constraints grows much faster for this model, as expected.

- The minimum cost flow model has the fewest variables and the itinerary-based model has the fewest constraints in all simulations except the fifth, which has a very small data set, where the itinerary-based model has less variables and less constraints.

Next we mention a few observations about the number of variables and constraints.

- The number of variables grows at similar rates for the itinerary-based and multi-commodity flow formulations.
  - It approximately doubles as P doubles.
  - It grows more slowly than expected as F increases and may be order $O(F^2P)$.

- For the minimum cost flow model, as P doubles, the number of variables less than doubles.

- For the minimum cost flow model, the number of variables increases less quickly than expected as F increases.

- For the itinerary-based and minimum cost flow models, the number of constraints grows at rates that are reasonably close, as predicted.

3.2 Stochastic Problem

Next, we consider the case where the departure time of the delayed flight is stochastic. There is a known, finite set of possible lengths for the delay, along with the probability of each possibility. The length of delay becomes known for certain shortly before the flight departs. The departure times for all other flights are known.

3.2.1 Approaches

We now consider a few options to solve this problem, with the goal of handling best the stochasticity. The current approach is the most difficult to implement but is the most beneficial to passengers. The other two are less difficult to implement and less helpful to passengers, but we discuss them briefly.

3.2.1.1 Current Approach

In the current chosen approach, we have a two-stage model. In the first stage, we assign an itinerary to each passenger as soon as the delay is discovered. Some passengers are given new itineraries that do not involve the delayed flight at all, and others may be assigned the original flight even if there is a possibility of missing a connection. Passengers may be assigned their original itineraries. Next, in the second stage problem, we evaluate the situation in each outcome for any passengers who will miss their connections with their first-stage itineraries and give those passengers new itineraries again.

Both first and second stages are solved together, where the costs for the second-stage problem in the different scenarios for the length of the delay are weighted by their probabilities. In solving the model, we decide what to do when the delay is discovered and what we will do in each scenario in the second stage if it occurs. Setting up and solving this model is the subject of the rest of this chapter and chapter IV.

We make a couple of rules for the passengers' itineraries. When a passenger has two or more flights in his or her original itinerary, the passenger can move to a different first flight in the first-stage problem only if the current itinerary has a chance of being disrupted. The passenger cannot move to a different itinerary simply because a one-flight itinerary would be more convenient, for example. A second situation where passengers can get a new itinerary in the first stage is when they have only the original flight and no other flights. Since that flight will be delayed, it makes sense to have the option of moving the passengers to other flights that are not delayed if their arrival time is earlier than the delayed flight's expected arrival time. These passengers do not necessarily receive the highest priority since they do not have connections to miss, but the goal is to get everyone to their destinations as early as possible.

Note that it is possible that a passenger can be given an itinerary in the first-stage problem that includes the delayed flight but is not the same as the passenger's original itinerary. This can be confusing to communicate to passengers. Thus, if a passenger is assigned a second flight that the passenger may miss, then it may be best to tell passengers what flights to take only after the delay becomes known and then the second-stage solution can be communicated to passengers. Now, before presenting the models, we discuss whether or not passengers that are assigned the delayed flight in the first stage are required to take that flight. We consider both possibilities.

- First, suppose there is no requirement that the passengers assigned to the original flight in stage one must wait for it, but they can instead be given new itineraries. This could be done if enough time has passed that they will miss their connections with their current stage-one itineraries. The second stage problem is then solved viewing the passengers as still in the original location. Using this method, less passengers will miss connections since some people assigned to the original flight in stage one will have switched to another flight. There are three different ways to put this first method into practice.

  - The first is to re-solve the problem as each possible length of delay is found not to occur. If a possible length of delay does not occur, the probabilities of each of the other later outcomes is greater than it was previously. The problem is then re-solved using these probabilities. This is good for passengers, but unfortunately, the problem becomes a several-stage problem and thus is much harder to solve. We will ignore this option in the rest of the dissertation.
  - Second, when an outcome is found not to happen, the passengers follow the solution for the next possible outcome. For example, if two of the possible outcomes for the delay are one hour and two hours, once the passengers have been waiting for at least one hour, they switch flights if that is what the solution says for a length of delay of two hours. This method gives just two stages.
  - Third, we say that passengers can switch from the original flight once the length of delay becomes known. For example, if the delay could be one, two, four, or six hours and the delay is found to be four hours, then the passengers may switch flights shortly before four hours have passed. This is also still a two-stage problem, but less convenient for passengers.

- In the second method, passengers that are assigned to the original delayed flight in stage one must take that flight no matter how long the delay turns out to be. Shortly before departure, all passengers assigned to the original flight board the aircraft and then it departs. Once the flight lands, the passengers who missed their connections are given their new second-stage itineraries. The second stage problem is solved viewing the passengers as having already flown on the original flight.

Each of the methods described in this section has its own advantages. The method where passengers assigned the original flight in the first stage do not necessarily have to take it is beneficial to the passengers. An advantages of the method where passengers assigned the original flight in stage one must take it no matter how long the delay turns out to be is that the problem is easier to understand and its solution is simpler to communicate to the passengers. The biggest advantage to using this method is that the solution found from solving the problem is still valid if the passengers have already boarded the plane, their luggage is on the plane, or there is any other reason it is not possible for passengers to switch from the original flight. Thus, we will only use this method in solving the two-stage problem, and we hope to investigate using the other methods in the future.

3.2.1.2 Other Approaches

Instead of using a two-stage problem, we could make the restriction that all connecting passengers must move to a new itinerary. There is no chance of passengers missing their connections, because only the original flight's departure time is stochastic. To do this, we let $P^{\star}$ be the set of all connecting passengers and change certain constraints in the different models.

- In the itinerary-based model, for all $p \in P^\star$, replace constraint (3.1), $\sum_{i \in I_p} x_{pi} = 1$, with $\sum_{i \in I_p : f_1 \notin F_i} x_{pi} = 1$ where flight f_1 is the delayed flight.
- In the flight-based model, let $u_a^p = 0$ for all $p \in P^\star$ where arc a is the arc going from the start node to the first flight node for the delayed flight.
- In the minimum cost flow model, make the capacity be 0 for all arcs going from the second flight node for the original flight to another flight node.

Second, we could say that any passenger can keep his or her original itinerary. If a connecting passenger keeps the original itinerary and could later miss the connection, make the cost for that itinerary be very high. For example, say that passenger p stays with the original itinerary and that the number of possible lengths for the delay is 4. The first three cases lead to him making his second flight, but with probability ρ_4, he will miss his connection. Then the cost c_{p14} in the last case of keeping the original itinerary is very high, which also makes $\sum_{j=1}^{4} \rho_j \cdot c_{p1j}$ very high, where c_{p1j} is the cost for passenger p to keep the original itinerary in case j for $j \in \{1, 2, 3, 4\}$.

These methods are more straightforward and easy to solve than the two-stage model. Unfortunately, in most situations, they cannot produce the best itineraries for some of the passengers. Depending on the possibilities for the length of delay and their probabilities, it is often best for passengers to keep their original itineraries even if there is a chance of missing connections.

3.2.2 Description of Models

We model the problem as a two-stage stochastic programming problem, as mentioned in section 3.2.1, where the goal is to minimize the sum of the delay for all passengers. We have a few different models in order to evaluate the advantages of each. The first is an itinerary-based model, the second is a flight-based model that is a multi-commodity flow formulation, the third is another flight-based model that is a minimum cost flow formulation, and the last is a combination of the itinerary-based and minimum cost flow formulations.

- The itinerary-based model has binary first and second-stage variables that represent giving a certain itinerary to a specific passenger. An advantage of this model is that it is easier to understand and is more concise than the others. Also, the objective function and the constraints are linear and both stages can be solved together. Note that since the second-stage variables are binary, we cannot use Benders decomposition as described in section 4.1.1. In addition, since the integrality constraints much be enforced, the method of branch and bound must be used, which can be quite time-consuming.

- In the first flight-based model, we have a network that represents passengers taking flights. We have a start node with a supply of 1 for each passenger that is disrupted, two flight nodes for each flight, and an end node with a demand of 1 for each disrupted passenger. Each variable represents the flow over an arc for a certain passenger. Two advantages of this model over the minimum cost flow model are that both stages can be solved together and the solution clearly shows which flights to give to each person. On the other hand, the objective function is not linear, which is difficult to overcome, since the simplex method cannot be used to solve the model. Also the variables are binary, so Benders decomposition cannot be used as explained in section 4.1.1.

- The network for the minimum cost flow formulation is very similar to that for the multi-commodity flow formulation, but all passengers are treated as one commodity and so each variable represents the total flow over an arc. This formulation is convenient to use since there are fewer variables than in the other three formulations, the objective function is linear, and there are not integrality constraints on the variables. The disadvantage is that it is not possible to solve both the first and second stage problems together as described in section 3.2.5. Also, after the variables are solved for, we have to solve a separate flow-to-path problem to decide what flights to give to each passenger.

- In the combined model, the first-stage problem is the same as that for the itinerary-based model, and the second-stage problem is a minimum cost flow model. Some advantages are that we have a linear objective function and the second-stage solutions are automatically integral, since the second-stage problem is a minimum cost flow model. Thus we can use Benders decomposition, which we will discuss in section 4.1. Also, the second-stage problem can be solved relatively quickly since the variables are continuous. Last, both stages can be solved together. This is the model that we implement in chapter IV.

The information received from solving any of these models is the itineraries to give to all the passengers when the delay or delays are discovered, which passengers will miss their connections in each scenario, and the new second-stage itineraries to give each of them in each scenario. When the problem has been solved, the airline gives the passengers their first-stage itineraries. The passengers that have been given the delayed flight must wait in the terminal for it until it departs, no matter how long that takes. The passengers who were not given the delayed flight must go find their new flights. Shortly before the delayed flight departs, the airline discovers the length of delay and the waiting passengers board the aircraft. After the flight lands, the airline can implement the solution found previously for the scenario that occurs and give the passengers who missed connections after taking the delayed flight their new itineraries.

A list and description of all of the sets, parameters, and variables for all four models are given in appendix A.

3.2.3 Itinerary-based Model

First, we have an itinerary-based model that builds on the deterministic one in section 3.1.1. The x_{pi} variables and the first-stage constraints, which are below, are the same as before.

$$\sum_{i \in I_p} x_{pi} = 1 \qquad \forall p \in P$$

$$\sum_{p \in P} \sum_{i \in I_p : f \in F_i} x_{pi} \leq C_f \qquad \forall f \in F$$

$$x_{pi} \in \{0,1\} \qquad \forall p \in P,\ \forall i \in I_p$$

Next, for the second-stage problem, we need some new notation.
- Let Ω be the set of possible outcomes for the length of the delay.
- Let ρ^{ω} be the probability of outcome $\omega \in \Omega$.
- Let ϵ_i^{ω} be 1 if itinerary i will get disrupted in outcome $\omega \in \Omega$. The only situation where this can happen is when the first flight in itinerary i is the original flight, since all other flight times are known. Otherwise let ϵ_i^{ω} be 0.
- Let I_p^{ω} be the set of possible itineraries in the second stage for passenger $p \in P$ if disrupted in outcome $\omega \in \Omega$. We determine this list in the same way that we determine I_p in section 3.1.1, except that the time is later and the location is the destination of the delayed flight since we choose to use the first method described for using a two-stage model in section 3.2.1. We can find I_p^{ω} easily since we know all the information about the flights that are scheduled to depart at each point in time.
- For $p \in P$, let the variable y_{pi}^{ω} be 1 if itinerary i is given to passenger p in stage two under outcome ω and let it be 0 otherwise.

The first set of second-stage constraints ensures that each passenger that will be disrupted with his or her first-stage itinerary in outcome $\omega \in \Omega$ is assigned exactly one new itinerary in that outcome and that no new itineraries are given to passengers who will not be disrupted in outcome ω. The constraints are

$$\sum_{i \in I_p^\omega} y_{pi}^\omega = \sum_{i \in I_p} \epsilon_i^\omega x_{pi}, \quad \forall p \in P, \, \omega \in \Omega \tag{3.12}$$

If a passenger p's itinerary $i \in I_p$ from stage one will not be disrupted in outcome ω, then ϵ_i^ω is 0, so the right hand side of the constraint is 0. Then y_{pi}^ω must be 0 for all possible itineraries $i \in I_p^\omega$ for passenger p under outcome ω, since that passenger will not need a new itinerary. If the passenger's itinerary from stage one will be disrupted in outcome ω, then ϵ_i^ω is 1, so the right hand side of the constraint is 1. Then the passenger must be given exactly one new itinerary, so the sum of y_{pi}^ω over all possible new itineraries i is 1.

The second set of constraints ensures that for each flight, the number of people assigned to the flight in outcome ω is no more than the available capacity after stage one. The constraints are

$$\sum_{p \in P} \sum_{i \in I_p^\omega : f \in F_i} y_{pi}^\omega \; \leq \; C_f - \sum_{p \in P} \sum_{i \in I_p : f \in F_i} (1 - \epsilon_i^\omega) x_{pi}, \quad \forall f \in F, \, \omega \in \Omega \tag{3.13}$$

The left side of each constraint is the number of people assigned to flight f in stage two under outcome ω. The right side is the available capacity after stage one. To see this, for each passenger p, if flight f is part of the itinerary given to the passenger in stage one, then x_{pi} is 1 for some $i \in I_p$ such that $f \in F_i$. If itinerary i will be disrupted, then ϵ_i^ω is 1, so $1 - \epsilon_i^\omega$ is 0, so $(1 - \epsilon_i^\omega)x_{pi}$ is 0, and the passenger did not actually take up a seat. If, on the other hand, itinerary i will not be disrupted, then ϵ_i^ω is 0, so $1 - \epsilon_i^\omega$ is 1. Then $(1 - \epsilon_i^\omega)x_{pi}$ is 1, so one seat is subtracted from the available capacity of the flight. Remember that if an itinerary is disrupted, the passengers on that itinerary miss all their flights except the original flight. Last, all y variables must be 0 or 1, so

$$y_{pi}^{\omega} \in \{0,1\}, \quad \forall p \in P,\, i \in I_p,\, \omega \in \Omega \tag{3.14}$$

Next, the objective function for stage one and stage two together is

$$\sum_{p\in P}\sum_{i\in I_p}\Big(1-\sum_{\omega\in\Omega}\epsilon_i^{\omega}\rho^{\omega}\Big)c_{pi}x_{pi}+\sum_{\omega\in\Omega}\rho^{\omega}\sum_{p\in P}\sum_{i\in I_p^{\omega}}c_{pi}y_{pi}^{\omega} \tag{3.15}$$

For all $p \in P$ and $i \in I_p$, c_{pi} is the cost of assigning itinerary i to passenger p in the first stage. If a passenger will be disrupted with the first-stage itinerary, the cost for the first stage should not be counted. In the objective function, we have $1 - \sum_{\omega\in\Omega} \epsilon_i^{\omega}\rho^{\omega}$ multiplied by $c_{pi}x_{pi}$. To see why, note that $\sum_{\omega\in\Omega} \epsilon_i^{\omega}\rho^{\omega}$ is the probability that itinerary i from stage one will be disrupted. Then $1 - \sum_{\omega\in\Omega} \epsilon_i^{\omega}\rho^{\omega}$ is the probability that itinerary i will not be disrupted, so the cost c_{pi} of itinerary i for passenger p is weighted by the probability that the cost will actually be counted. For the second term, $\sum_{p\in P}\sum_{i\in I_p^{\omega}} c_{pi}y_{pi}^{\omega}$ is the total cost for the second stage under outcome ω. We weight each outcome by it probability ρ^{ω} where $\sum_{\omega\in\Omega}\rho^{\omega} = 1$, and arrive at the expression above.

Consider a passenger p whose original itinerary consists of only the delayed flight. Suppose that in the first stage, this passenger is given the same itinerary, so that $x_{p1} = 1$ where itinerary 1 has only the delayed flight. Unlike the other first-stage itineraries, the arrival time is not deterministic. Thus, we let c_{p1} be the expected cost, $\sum_{\omega\in\Omega}\rho^{\omega}d^{\omega}$, where d^{ω} is the length of delay for the delayed flight in outcome $\omega \in \Omega$.

3.2.4 Flight-based Model

We next look at the two-stage model that builds on the flight-based model from section 3.1.2. The x_a^p variables and the first-stage constraints, below, are defined as before in section 3.1.2.

$$
\begin{aligned}
&\sum_{a \in B_n} x_a^p - \sum_{a \in E_n} x_a^p = s_n^p && \forall n \in N,\, p \in P \\
&x_a^p \leq u_a^p && \forall a \in A_e,\, p \in P \\
&\sum_{p \in P} x_a^p \leq u_a && \forall a \in A_{fp} \\
&x_a^p \in \{0, 1\} && \forall a \in A,\, p \in P
\end{aligned}
$$

In stage two, for each $\omega \in \Omega$, there are a couple of changes to the network from the first stage.

- First, the start and end nodes have different supplies than in stage one. There is a supply of 1 at the start node and a demand of 1 at the end node for each passenger who will be disrupted with his or her stage-one itinerary in outcome ω. These are both 0 for all non-disrupted passengers.

- Next, the arcs originating at the start node are different. The people disrupted after stage one are now in a new location, and time has passed by stage two, so some flights are no longer available. The information about these flights is known.

For our formulation, we need some new sets, parameters, and variables. First, let A_c^{s1} be the set of arcs in the first-stage network that begin at the second flight node of the delayed flight and connect to the first flight node for another flight. These arcs represent having a connecting flight from the original flight. Next, we have a few new sets and parameters that are common to all the stage-two subproblems.

Let Ω be the set of possible outcomes for the length of the delay.

Let N_{fd} be the set of the first flight nodes from a pair for flights that begin at the destination of the original flight.

Let B_n be the set of arcs that begin at any node $n \in N$ except the start node.

Let E_n be the set of arcs that end at node $n \in N \setminus N_{fd}$.

Let n_1 be the start node.

Let n_l be the end node.

We also have a few new sets, parameters, and variables that apply to each outcome $\omega \in \Omega$.

Let A^{ω} be all the arcs in the network.

Let B_1^{ω} be the set of arcs that begin at the start node.

Let E_n^{ω} be the set of arcs that end at node $n \in N_{fd}$.

Let ϵ_a^{ω} be 1 if using arc $a \in A_c^{s1}$ is not possible and let it be 0 otherwise.

Let $y_a^{p\omega}$ be 1 if arc a is used for passenger p and 0 otherwise, for all $p \in P$ and $a \in A^{\omega}$.

Next, we look at the constraints in the model for the second stage for each outcome $\omega \in \Omega$. For each passenger, we want node n_1 to have a supply of 0 if the passenger will reach the destination with the itinerary from stage one and a supply of 1 if the passenger will miss a connection. Thus let the supply at the start and end node be $s_{n_1}^{p\omega} = \sum\limits_{a \in A_c^{s1}} \epsilon_a^{\omega} x_a^p$, and $s_{n_l}^{p\omega} = -s_{n_1}^{p\omega}$, respectively. To see this, if $\epsilon_a^{\omega} = 1$, then the original flight will be so late that passengers will not be able to connect to the flight at the end of arc $a \in A_c^{s1}$. Note that the value of ϵ_a^{ω} is known ahead of time for each ω. A passenger p will be disrupted under outcome ω if $\sum\limits_{a \in A_c^{s1}} \epsilon_a^{\omega} x_a^p$ is 1 and will not be disrupted if it is 0. The supply at all nodes other than the start and end nodes is 0. Thus the balance constraints are

$$\sum_{a \in B_1^{\omega}} y_a^{p\omega} = \sum_{a \in A_c^{s1}} \epsilon_a^{\omega} x_a^p, \quad \forall p \in P, \, \omega \in \Omega \tag{3.16}$$

$$\sum_{a \in E_{n_l}} y_a^{p\omega} = \sum_{a \in A_c^{s1}} \epsilon_a^{\omega} x_a^p, \quad \forall p \in P,\, \omega \in \Omega \tag{3.17}$$

$$\sum_{a \in B_n} y_a^{p\omega} - \sum_{a \in E_n^{\omega}} y_a^{p\omega} = 0, \quad \forall n \in N \setminus \{n_1, n_l\}, \forall p \in P,\, \omega \in \Omega \tag{3.18}$$

The only arcs with commodity-specific capacities are those in A_e. The capacities enforce the rule that a passenger can go to the end node only by arriving at his or her desired destination. These capacities do not change from stage one to stage two, so

$$y_a^{p\omega} \leq u_a^p, \quad \forall a \in A_e,\, \omega \in \Omega \tag{3.19}$$

Next, for $a \in A_{fp}$, which are the arcs between two flight nodes in a pair, the capacity on a is the available capacity before stage one minus the number of passengers who will use seats on the first-stage flights. For each passenger p, recall that $\sum\limits_{b \in A_c^{s1}} \epsilon_b^{\omega} x_b^p$ is 1 if the passenger will be disrupted under outcome ω and is 0 otherwise. Then $1 - \sum\limits_{b \in A_c^{s1}} \epsilon_b^{\omega} x_b^p$ is 0 if the passenger will be disrupted and is 1 otherwise. Then for any arc $a \in A_{fp}$, $\Big(1 - \sum\limits_{b \in A_c^{s1}} \epsilon_b^{\omega} x_b^p\Big) x_a^p$ is 0 if passenger p will not use arc a in stage one under outcome ω, and is 1 if the person will use arc a. Thus $\sum\limits_{p \in P}\Big(1 - \sum\limits_{b \in A_c^{s1}} \epsilon_b^{\omega} x_b^p\Big) x_a^p$ is the number of passengers that will use arc a in stage one. For each $a \in A_{fp}$, we have that

$$\sum_{p \in P} y_a^{p\omega} \leq u_a - \sum_{p \in P}\Big(1 - \sum_{b \in A_c^{s1}} \epsilon_b^{\omega} x_b^p\Big) x_a^p = u_a + \sum_{p \in P}\Big(\sum_{b \in A_c^{s1}} \epsilon_b^{\omega} x_b^p - 1\Big) x_a^p \tag{3.20}$$

Last, each arc is either used or not used for each passenger, so we need to add the constraints

$$y_a^{p\omega} \in \{0, 1\}, \quad \forall a \in A,\, p \in P,\, \omega \in \Omega \tag{3.21}$$

Consider the objective function for the problem. Letting ρ^ω be the probability of outcome $\omega \in \Omega$, at first it seems as if the objective function should be $\displaystyle\sum_{p\in P}\sum_{a\in A_e} c_a^p x_a^p + \sum_{\omega\in\Omega}\rho^\omega \sum_{p\in P}\sum_{a\in A_e} c_a^p y_a^{p\omega}$. If a passenger will be disrupted though, then the cost in the first stage should not be counted. For each passenger p, we want to subtract $\displaystyle\sum_{a\in A_e} c_a^p x_a^p \cdot \sum_{a\in A_c^{s1}} \epsilon_a^\omega x_a^p$ from the total cost. Then the objective function becomes

$$\sum_{p\in P}\sum_{a\in A_e} c_a^p x_a^p + \sum_{\omega\in\Omega}\rho^\omega \left(\sum_{p\in P}\sum_{a\in A_e} c_a^p y_a^{p\omega} - \sum_{p\in P}\sum_{a\in A_e} c_a^p x_a^p \cdot \sum_{a\in A_c^{s1}} \epsilon_a^\omega x_a^p\right) \tag{3.22}$$

which equals

$$\sum_{p\in P}\sum_{a\in A_e} c_a^p x_a^p \left(1 - \sum_{\omega\in\Omega}\rho^\omega \sum_{a\in A_c^{s1}} \epsilon_a^\omega x_a^p\right) + \sum_{\omega\in\Omega}\rho^\omega \sum_{p\in P}\sum_{a\in A_e} c_a^p y_a^{p\omega} \tag{3.23}$$

To see why this makes sense, $\displaystyle\sum_{a\in A_c^{s1}} \epsilon_a^\omega x_a^p$ is 1 if passenger p will be disrupted with his or her first-stage itinerary in outcome ω and is 0 otherwise. Then $\displaystyle 1 - \sum_{\omega\in\Omega}\rho^\omega \sum_{p\in P}\sum_{a\in A_c^{s1}} \epsilon_a^\omega x_a^p$ is the probability that passenger p will not be disrupted, and the cost in the first stage is weighted by this probability.

3.2.5 Minimum Cost Flow Model

We can also explore the idea of using a two-stage problem with the minimum cost flow model in section 3.1.3. Our minimum cost flow formulation is very similar to the multi-commodity flow formulation in section 3.2.4. The first and second-stage networks and the notation have much in common. The difference here is that all passengers are treated as one commodity, instead of each passenger being his or her own commodity. Some of the similarities and differences in the network and notation are described at the beginning of section 3.1.3, and the rest are given below.

- In the second-stage problem, we have an end node for each passenger that is disrupted in outcome ω. Those passengers' end nodes each has a demand of 1.
- Let N_p be the set of end nodes.
- Let $n_p \in N_p$ be the end node corresponding to passenger $p \in P$.
- Let δ_a^ω be the number of passengers who are supposed to use arc $a \in A_e \cup A_{fp}$ in stage one but will not due to misconnections. This is not possible to express in the model in term of the x variables, but it could be computed off-line.
- Let δ_{ap}^ω be the number of units of flow that are supposed to be sent over arc $a \in A_c^{s1}$ to end at the end node for passenger $p \in P$ with the first-stage solution but will not because of misconnections. This also is not possible to express in the model in term of the x variables, but it could be computed off-line.
- Let y_a^ω be the number of units of flow (passengers) sent over arc $a \in A^\omega$.

The x_a variables and the first-stage constraints, which are below, are the same as before.

$$\sum_{a \in B_n} x_a - \sum_{a \in E_n} x_a = s_n \qquad \forall n \in N$$

$$x_a \leq u_a \qquad \forall a \in A_{fp}$$

$$x_a \geq 0 \qquad \forall a \in A$$

Next, the supply at the start node in the second stage under outcome ω is $\sum\limits_{a \in A_c^{s1}} \epsilon_a^\omega x_a$ and the supply at the end node corresponding to passenger $p \in P$ is $\sum\limits_{a \in A_c^{s1}} \delta_{ap}^\omega$. For all nodes other than the start node and those in N_p, the supply is 0. Thus balance constraints for each $\omega \in \Omega$ are

$$\sum_{a \in B_1^\omega} y_a^\omega = \sum_{a \in A_c^{s1}} \epsilon_a^\omega x_a, \quad \forall \omega \in \Omega \tag{3.24}$$

$$\sum_{a \in E_{n_p}} y_a^\omega = \sum_{a \in A_c^{s1}} \delta_{ap}^\omega, \quad \forall n_p \in N_p,\, \omega \in \Omega \tag{3.25}$$

$$\sum_{a \in B_n} y_a^\omega - \sum_{a \in E_n^\omega} y_a^\omega = 0, \quad \forall n \in N \setminus (\{n_1\} \cup N_p),\, \omega \in \Omega \tag{3.26}$$

The second set of constraints is

$$y_a^\omega \leq u_a - x_a + \delta_a^\omega, \quad \forall a \in A_{fp},\, \omega \in \Omega \tag{3.27}$$

which ensures that the available flight capacities are not violated. Last, for all $a \in A^\omega$, $\omega \in \Omega$, we have that $y_a^\omega \in \{0, 1\}$, which becomes

$$y_a^\omega \geq 0, \quad \forall a \in A^\omega,\, \omega \in \Omega \tag{3.28}$$

since this is a minimum cost flow formulation with integer supplies and capacities.

Next, consider the objective function. The cost for the first stage is $\sum\limits_{a \in A_e} c_a x_a$, and the cost for the second stage is $\sum\limits_{\omega \in \Omega} \rho^\omega \sum\limits_{a \in A_e} c_a y_a^\omega$. The objective function is just the sum of these two expressions, except that we need to subtract the cost in stage one for any passengers who will not reach their destinations with their stage-one itineraries. Since for all $a \in A_e$, δ_a^ω is the number of passengers who are supposed to use arc a in stage one and will not and ρ^ω is the probability of outcome $\omega \in \Omega$, then $\sum\limits_{\omega \in \Omega} \rho^\omega \delta_a^\omega$ is the expected number of passengers that plan to use arc $a \in A_e$ but will not in stage one. We want to subtract this amount from the flow over each arc in $a \in A_e$. Thus, the objective function for both the first and second stages is

$$\sum_{a \in A_e} c_a(x_a - \sum_{\omega \in \Omega} \rho^\omega \delta_a^\omega) + \sum_{\omega \in \Omega} \rho^\omega \sum_{a \in A_e} c_a y_a^\omega \tag{3.29}$$

The solution to this model provides us with the amount of flow on every arc in the first-stage problem and in each scenario in the second-stage problem. How can we determine what flights to assign to each passenger from this information? We can trace the flows backward from the end nodes in each stage, as described at the end of section 3.1.3 and in section 3.2.7. Also in section 3.2.7, we prove that the path-based solution found from a given arc-based solution is unique, and we give an algorithm to find the path-based solution.

3.2.6 Combined Model

Next, we have a combined model. We use the itinerary-based model for the first stage and the minimum cost flow model for the second stage. We may have to use the branch and bound method for the first-stage problem since the itinerary-based formulation does not automatically have integer solutions without enforcing the integrality constraints.

First, we have some new notation. Let $A_{fpi} \in A_{fp}$ be the set of arcs in the minimum cost flow model that correspond to itinerary $i \in I_p$ for all $p \in P$ in stage one in the itinerary-based formulation.

The first part of the objective function and the first two constraints are the same as in section 3.2.3. The second part of the objective function and the fifth constraint are the same as in section 3.2.5. The third and fourth constraints combine the previous two models and cause each passenger to get a new itinerary in the second stage if and only if he or she will be disrupted in the first stage. The sixth constraint,

$$y_a^\omega \leq u_a - \sum_{p \in P} \sum_{i \in I_p : a \in A_{fpi}} (1 - \epsilon_i^\omega) x_{pi}, \quad \forall a \in A_{fp},\, \omega \in \Omega \tag{3.30}$$

also combines the two previous models and ensures that the available capacity in the second stage is not exceeded, taking into account the passengers that will take up seats based on the first-stage decisions. To see this, the left hand side of the equation is the number of passengers who will use arc $a \in A_{fp}$ in stage two under outcome ω. On the right side, for arc a and a passenger $p \in P$, $\sum\limits_{i \in I_p : a \in A_{fpi}} (1 - \epsilon_i^\omega) x_{pi}$ is 1 if the passenger plans to use the arc in stage one and will not be disrupted, and it is 0 if the passenger either plans to use the arc in stage one and will be disrupted so will not take up a seat or does not plan to use the arc at all. Thus the new model is

$$\min \sum_{p \in P} \sum_{i \in I_p} \Big(1 - \sum_{\omega \in \Omega} \epsilon_i^\omega \rho^\omega\Big) c_{pi} x_{pi} + \sum_{\omega \in \Omega} \rho^\omega \sum_{a \in A_e} c_a y_a^\omega \tag{3.31}$$

$$\text{s.t.} \quad \sum_{i \in I_p} x_{pi} = 1 \qquad \forall p \in P \tag{3.32}$$

$$\sum_{p \in P} \sum_{i \in I_p : f \in F_i} x_{pi} \leq C_f \qquad \forall f \in F \tag{3.33}$$

$$\sum_{a \in B_1^\omega} y_a^\omega = \sum_{p \in P} \sum_{i \in I_p} \epsilon_i^\omega x_{pi} \qquad \forall \omega \in \Omega \tag{3.34}$$

$$\sum_{a \in E_{n_p}} y_a^\omega = \sum_{i \in I_p} \epsilon_i^\omega x_{pi} \qquad \forall p \in P,\ \omega \in \Omega \tag{3.35}$$

$$\sum_{a \in B_n} y_a^\omega - \sum_{a \in E_n^\omega} y_a^\omega = 0 \qquad \forall n \in N \setminus (\{n_1\} \cup N_p),\ \omega \in \Omega \tag{3.36}$$

$$y_a^\omega \leq u_a - \sum_{p \in P} \sum_{i \in I_p : a \in A_{fpi}} (1 - \epsilon_i^\omega) x_{pi} \qquad \forall a \in A_{fp},\ \omega \in \Omega \tag{3.37}$$

$$x_{pi} \in \{0, 1\} \qquad \forall p \in P,\ \forall i \in I_p \tag{3.38}$$

$$y_a^\omega \geq 0 \qquad \forall \omega \in \Omega,\ a \in A^\omega \tag{3.39}$$

In this model, the first-stage problem is

$$\begin{aligned}
\min\ & \sum_{p \in P} \sum_{i \in I_p} \Big(1 - \sum_{\omega \in \Omega} \epsilon_i^{\omega} \rho^{\omega}\Big) c_{pi} x_{pi} && && (3.40) \\
\text{s.t. } & \sum_{i \in I_p} x_{pi} = 1 && \forall p \in P && (3.41) \\
& \sum_{p \in P} \sum_{i \in I_p : f \in F_i} x_{pi} \leq C_f && \forall f \in F && (3.42) \\
& x_{pi} \in \{0, 1\} && \forall p \in P,\ \forall i \in I_p && (3.43)
\end{aligned}$$

and the second-stage problem for each $\omega \in \Omega$ is

$$\begin{aligned}
\min\ & \sum_{a \in A_e} c_a y_a^{\omega} && && (3.44) \\
\text{s.t. } & \sum_{a \in B_1^{\omega}} y_a^{\omega} = \sum_{p \in P} \sum_{i \in I_p} \epsilon_i^{\omega} x_{pi} && && (3.45) \\
& \sum_{a \in E_{n_p}} y_a^{\omega} = \sum_{i \in I_p} \epsilon_i^{\omega} x_{pi} && \forall p \in P && (3.46) \\
& \sum_{a \in B_n} y_a^{\omega} - \sum_{a \in E_n^{\omega}} y_a^{\omega} = 0 && \forall n \in N \setminus (\{n_1\} \cup N_p) && (3.47) \\
& y_a^{\omega} \leq u_a - \sum_{p \in P} \sum_{i \in I_p : a \in A_{fpi}} (1 - \epsilon_i^{\omega}) x_{pi} && \forall a \in A_{fp} && (3.48) \\
& y_a^{\omega} \geq 0 && \forall\ a \in A^{\omega} && (3.49)
\end{aligned}$$

For the second-stage problem, a solution to this model gives us the amount of flow on every arc in the network. We next have to determine what flights to assign to each passenger from this information, so we need to find the path-based solution from the arc-based solution that we found. We can trace the flows backward from the end nodes, as described in section 3.1.3 and 3.2.7. Also in section 3.2.7, we prove that the path-based solution found from a given arc-based solution is unique, and we give an algorithm to find the path-based solution.

3.2.7 Arc-based to Path-based Solutions

Whenever we have solved one of our minimum cost flow models from section 3.1.3, 3.2.5, or 3.2.6, the solution is the flow over all the arcs in the corresponding network. We need to be able to convert this solution into information telling us what itinerary to give to each passenger. Thus, we need to convert our arc-based solution into a path-based solution. We can think of this problem as finding a path from each passenger p's end node n_p back to the start node n_s for all $p \in P$, based on the solution for the flow over all arcs. Those paths make up the path-based solution.

It turns out that there is a unique path-based solution given an arc-based solution to one of our minimum cost flow models when the arc-based solution is a basic feasible solution. This is a valuable property, since otherwise the algorithm to convert from the arc-based to path-based solution would be rather complicated since we would have to deal with decisions at certain nodes. Following is a proof of this property.

Proof. Suppose we have an arc-based solution. Since we use Simplex, it is a basic feasible solution. By Theorem 7.4 in *Bertsimas and Tsitsiklis* (1997), that solution is a tree solution. By Theorem 7.1c in *Bertsimas and Tsitsiklis* (1997), there is a unique path from any node to any other node in the corresponding network. To see this, suppose that there are two different paths between two nodes n_1 and n_2. Join these two paths and delete any arcs that are part of both of them. This gives a cycle, which is a contradiction to the definition of a tree. Thus, there is a unique path from any passenger's end node to the start node. Thus, there is a unique path-based solution for our arc-based solution. □

Once we have solved for our arc variables, it is not hard to find the path-based solution to assign to each passenger, because there is only one set of flights within the arc-based solution that will take a passenger from the start node to his or her destination. We start at one passenger's end node, and follow the unique positive flow back to the start node, then repeat for each of the other passengers. We do not have to make any choices along the way. Next we have an algorithm, along with some notation, that describes our simple method of finding the path-based solution from a given arc-based solution. We write it as if the arc-based solution is in terms of variables $\mathbf{x}$, but the algorithm is the same if the variables are $\mathbf{y}^{\omega}$.

Let A_p be the arcs in the path chosen for passenger $p \in P$.
Let $N_{s,adj}$ be the set of nodes connected to the start node by an arc a where $x_a > 0$.
Let $a_{e,n}$ be the arc that ends at node $n \in N$ where $x_n > 0$.
Let n_p be the end node for passenger $p \in P$.
Let n^b_a be the node at the beginning of arc a.
Let n_c be the current node at any point in the algorithm.

```
for ($p = 1..|P|$) {
    $A_p = \emptyset$
    $n_c = n_p$
    while ($n_c \notin N_{s,adj}$) {
        $a^\star = a_{e,n_c}$
        Add arc $a^\star$ to $A_p$
        $n_c = n^b_{a^\star}$
    }
    $a^\star = a_{e,n_c}$
    Add arc $a^\star$ to $A_p$
    Subtract 1 from the flow on all arcs in $A_p$
}
```

Now, we prove that this algorithm will always work.

Proof. Suppose that for some passenger $p \in P$, when finding that passenger's path from the end node n_p to the start node n_s by using the algorithm, we arrive at a node $n_c \neq n_s$ where there is no positive flow coming into it that has not already been used for another passenger. There is then no way to get to node n_s.
Case 1: $n_c = n_p$, so there is no flow coming into node n_p. Then the demand of 1 is not satisfied there, which is a contradiction.
Case 2: $n_c \in N \setminus (N_p \cup \{n_s\})$. Then n_c is a transshipment node, and the flow out of node n_c is greater than the flow into node n_c, which is a contradiction.
Thus, the algorithm will always produce a path for each passenger from his or her end node back to the start node. □

We next illustrate the algorithm with an example. Suppose that a network model has the solution shown in figure 3.3. Note that it is a basic feasible solution since there are 17 arcs and 18 balance constraints, of which 17 are linearly independent. Also, it is a tree solution as expected. To find the path-based solution, we start at the end node for passenger $p = 1$. We use arcs a_{11}, a_4, and a_1 to arrive at the start node, then subtract 1 from the flow on each of those arcs. This gives the network shown in figure 3.4. We repeat this process for passengers $p = 2$ through $p = 7$, as shown in figures 3.5 through 3.10. Note that at no point does a choice have to be made between two or more arcs with positive flow remaining.

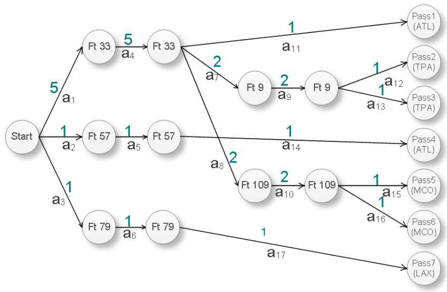

Figure 3.3: Solution

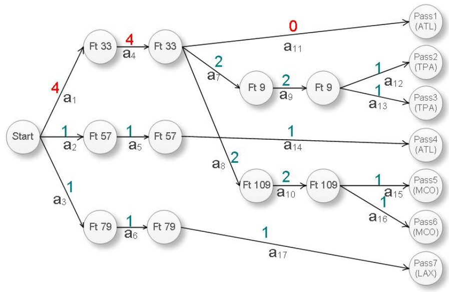

Figure 3.4: $A_1 = \{a_{11}, a_4, a_1\}$

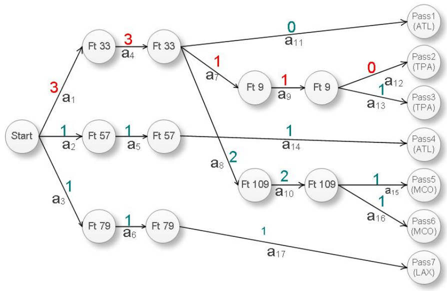

Figure 3.5: $A_2 = \{a_{12}, a_9, a_7, a_4, a_1\}$

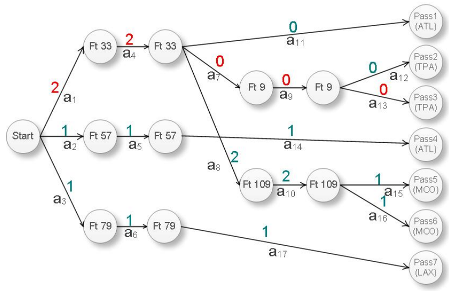

Figure 3.6: $A_3 = \{a_{13}, a_9, a_7, a_4, a_1\}$

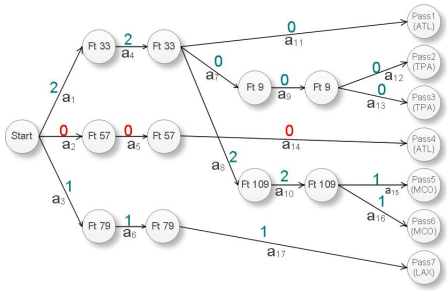

Figure 3.7: $A_4 = \{a_{14}, a_5, a_2\}$

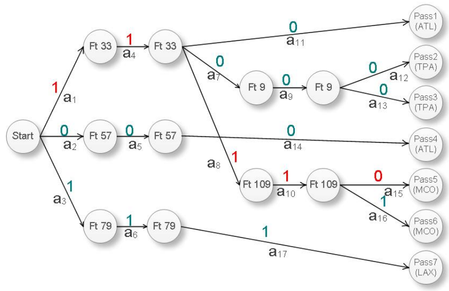

Figure 3.8: $A_5 = \{a_{15}, a_{10}, a_8, a_4, a_1\}$

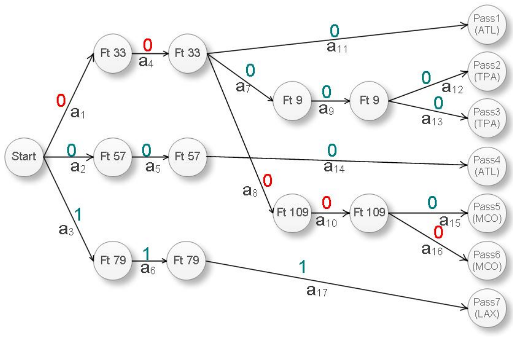

Figure 3.9: $A_6 = \{a_{16}, a_{10}, a_8, a_4, a_1\}$

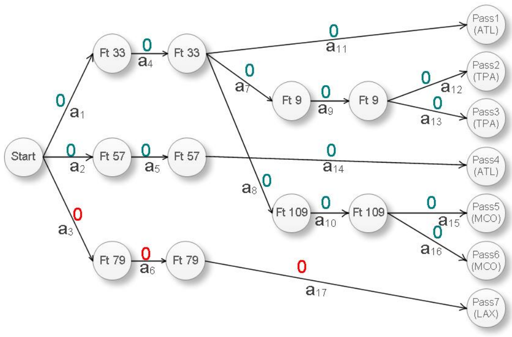

Figure 3.10: $A_7 = \{a_{17}, a_6, a_3\}$

3.2.8 Applications to Other Areas of Research

The results from our work may affect other areas of research as well, not just airline recovery. The results could apply to many situations where the flow on one arc in a network is stochastic, especially in a multi-commodity flow problem. It could apply to many recovery situation and planning problems. First, other recovery situations include other topics in airline recovery. We could use similar techniques as we have for passengers to improve recovery for aircraft, crew, and pilots. We could preemptively reroute them and also consider possibilities for lengths of delay while making decisions. Other areas of recovery could include rerouting traffic after an accident or during road construction, rerouting trains and buses when schedules get changed for some reason, like hazardous driving conditions or late drivers. The recovery could even apply to rescheduling doctors and nurses in hospitals.

The techniques could also be used in many different planning situations. For example, suppose people are planning the construction of a new freeway system where the amount of traffic at different times of the day is easy to estimate for most locations but traffic at a particular area or time is unknown. Working on how to handle disrupted flights could help make decisions such as how many lanes to put in the road, where exactly to construct it, and how to handle traffic lights. Last, our research could contribute to work such as that in *Applegate et al.* (2004), where algorithms are developed to find optimal restoration paths after node or link failures in a network.

CHAPTER IV

Implementation

As mentioned in chapter III, we have implemented the combined model in section 3.2.6, since it has a linear objective function and constraints, and the second-stage solution is automatically integral since it is a minimum cost flow model. In this chapter, we first introduce and discuss the method of Benders decomposition, a popular method used to decrease the solve time of large models. We derive it, using the explanation in *Bertsimas and Tsitsiklis* (1997), explain how our model can be solved using it, do an example by hand, and discuss some previous research using Benders decomposition to help solve airline planning and recovery problem.

Next, we present computational results on the solutions to our model in different situations, where the data set and possible lengths of delay change. In particular, we compare the delay experienced by the passengers from using our method versus reaccommodating the passengers one-by-one after misconnections have occurred. We discuss our data used, describing the data set and its size and talking about how it affects our model. Last, the run time in solving the model in different situations is next evaluated, and we specifically discuss how using Benders decomposition affects the run time with different sizes of the data set.

4.1 Benders Decomposition

4.1.1 Description

Benders decomposition is a method often used for large linear programming problems with a special structure. The objective function is a sum of terms involving one set of variables $\mathbf{x}$ and terms involving another set of variables $\mathbf{y}$. There are some constraints involving only the $\mathbf{x}$ variables and some constraints that involve only $\mathbf{y}$ or $\mathbf{x}$ and $\mathbf{y}$. Also, it is important that the $\mathbf{y}$ variables are all continuous. Thus they can not have any integrality requirements. The reason for this will be explained more below. The purpose of using Benders decomposition is to reduce the problem into several smaller problems that are easier to solve in order to make solving the complete model faster.

In particular, Benders decomposition can be used for two-stage stochastic linear programming problems. In the first stage, some decisions are made, determining the $\mathbf{x}$ variables, and then more information becomes available. There is a set Ω of finite size of known possibilities for the outcome ω, of this information. Each outcome ω has a probability ρ^ω of occurring. Unfortunately, ω is not known at the time $\mathbf{x}$ is chosen, but the information about ω is used in the second stage to make another set of decisions and determine the $\mathbf{y}^\omega$ variables for whichever ω occurs. Now choosing $\mathbf{x}$ and then $\mathbf{y}^\omega$ may not produce an optimal solution. Thus, a problem involving the $\mathbf{x}$ variables and $\mathbf{y}^\omega$ variables for all $\omega \in \Omega$ is set up, taking into account the probability of each possible outcome, and the goal is to solve for all variables simultaneously.

To use Benders decomposition, we first ignore all the constraints involving $\mathbf{y}$ and solve for $\mathbf{x}$. Then $\mathbf{x}$ is considered fixed and we want to find $\mathbf{y}$ by solving a subproblem. The objective function is the part of the original objective function that involves $\mathbf{y}$, and the constraints are the ones in the original problem that involve $\mathbf{y}$. From solving the subproblem, we get constraints on $\mathbf{x}$ if the $\mathbf{x}$ values previously chosen are not optimal once the second stage is taken into account or they make the subproblem infeasible. We add these constraints back into the original problem, solve for $\mathbf{x}$ again, and keep up this process until no constraints on $\mathbf{x}$ are found when solving the subproblems. Then the solution has been found.

Now, in each subproblem, the right hand side of the constraints involves $\mathbf{x}$, so each time we re-solve it, the extreme points of the corresponding polyhedron change. Thus we instead form the dual of the subproblem and use it to solve for $\mathbf{y}$ each time, since the dual has $\mathbf{x}$ in the objective function instead of in the right hand side of the constraints.

4.1.2 Derivation

The original problem is

$$\begin{aligned} \text{minimize}\quad & \mathbf{c}'\mathbf{x}+\mathbf{d}'\mathbf{y} \\ \text{subject to}\quad & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \mathbf{E}\mathbf{y} - \mathbf{F}\mathbf{x} = \mathbf{g} \\ & \mathbf{x}, \mathbf{y} \geq \mathbf{0} \end{aligned} \tag{4.1}$$

Here, the decision variables are $\mathbf{x}$ and $\mathbf{y}$ and are of size n_1 and n_2. The inputs in the problem are the vectors $\mathbf{c}$, $\mathbf{d}$, $\mathbf{b}$, and $\mathbf{g}$ of length n_1, n_2, m_1, and m_2, respectively, and the matrices $\mathbf{A}$, $\mathbf{E}$, and $\mathbf{F}$ of size $m_1 \times n_1$, $m_2 \times n_2$, and $m_2 \times n_1$, respectively. If it is a two-stage problem, then the first and second stage problems are

$$\begin{aligned} \text{minimize}\quad & \mathbf{c}'\mathbf{x} \\ \text{subject to}\quad & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{aligned} \tag{4.2}$$

$$\begin{aligned} \text{minimize} \quad & \mathbf{d}'\mathbf{y} \\ \text{subject to} \quad & \mathbf{E}\mathbf{y} - \mathbf{F}\mathbf{x} = \mathbf{g} \\ & \mathbf{y} \geq \mathbf{0} \end{aligned}$$

We can re-write model (4.1) in the following form:

$$\begin{aligned} \text{minimize} \quad & \mathbf{c}'\mathbf{x} + \theta \\ \text{subject to} \quad & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \theta = \min\{\mathbf{d}'\mathbf{y} \,|\, \mathbf{E}\mathbf{y} - \mathbf{F}\mathbf{x} = \mathbf{g}, \mathbf{y} \geq \mathbf{0}\} \\ & \mathbf{x} \geq \mathbf{0} \end{aligned}$$

Next, we have assumed that the $\mathbf{y}$ variables are continuous. Strong duality then tells us that if the primal subproblem (4.2) has an optimal solution, then the dual subproblem does also, and the optimal costs are the same (*Bertsimas and Tsitsiklis* (1997)). Note that if the $\mathbf{y}$ variables were not continuous, which would be true if there were integrality constraints, then strong duality would not hold. Instead we would have weak duality, which says that if $\mathbf{y}$ and $\mathbf{p}$ are any feasible solutions to the primal and dual problems, respectively, then the cost of $\mathbf{p}$ is less than or equal to the cost of $\mathbf{y}$ (*Bertsimas and Tsitsiklis* (1997)). Weak duality would not guarantee correct solutions when using Benders decomposition. Note that the $\mathbf{x}$ variables may have integrality requirements while Benders decomposition is being used. Thus, we can use the dual for the subproblem and re-write problem (4.1) as

$$\begin{aligned} \text{minimize} \quad & \mathbf{c}'\mathbf{x} + \theta \\ \text{subject to} \quad & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \theta = \max\{\mathbf{p}'(\mathbf{F}\mathbf{x} + \mathbf{g}) \,|\, \mathbf{p}'\mathbf{E} \leq \mathbf{d}'\} \\ & \mathbf{x} \geq \mathbf{0} \end{aligned}$$

The decision variables for the dual subproblem are $\mathbf{p}$ of length m_2, and the inputs are the vectors $\mathbf{g}$ and $\mathbf{f}$ and the matrices $\mathbf{F}$ and $\mathbf{E}$.

Next, the feasible region of a linear program is always a polyhedron, a set $P = \{\mathbf{z} \in \Re^n \mid \mathbf{C}\mathbf{z} \geq \mathbf{h}\}$, where $\mathbf{C}$ is an $m \times n$ matrix, $\mathbf{h} \in \Re^m$, and $\mathbf{z}$ are the n decision variables. Also, an extreme point is a vector $\mathbf{z} \in P$ if there does not exist $\mathbf{z}_1, \mathbf{z}_2 \in P$ and $\lambda \in [0, 1]$, where $\mathbf{z}_1, \mathbf{z}_2 \neq \mathbf{z}$ and $\mathbf{z} = \lambda\mathbf{z}_1 + (1 - \lambda)\mathbf{z}_2$. Last, if a linear program has an optimal solution and its feasible region has an extreme point, then it has an optimal solution which is an extreme point (*Bertsimas and Tsitsiklis* (1997)). Thus we know that our dual subproblem is optimized at an extreme point, so $\theta = \max\limits_i\{\mathbf{q}_i'(\mathbf{F}\mathbf{x} + \mathbf{g})\}$ for $\mathbf{q}_i \in Q$, the set of extreme points of the feasible region of the dual subproblem, which is the same as saying that θ is the smallest number where $\theta \geq (\mathbf{q}_i)'(\mathbf{F}\mathbf{x} + \mathbf{g})$ for all $\mathbf{q}_i \in Q$. Note that the size of an extreme point $\mathbf{q}_i \in Q$ is the same as the size of $\mathbf{p}$, which is m_2. Also, let W be the set of extreme rays of the dual subproblem. If the dual subproblem is unbounded, which also means that the primal subproblem is infeasible, then there exists an extreme ray $\mathbf{w}_i \in W$ of size m_2 where $\mathbf{w}_i'(\mathbf{F}\mathbf{x} + \mathbf{g}) > 0$. Thus, we add the constraints $\mathbf{w}_i'(\mathbf{F}\mathbf{x} + \mathbf{g}) \leq \mathbf{0}$ for all $\mathbf{w}_i \in W$. Thus, we can re-write the problem (4.1) in the next two formulations.

$$\begin{array}{ll}
\text{minimize} & \mathbf{c}'\mathbf{x} + \theta \\
\text{subject to} & \mathbf{A}\mathbf{x} = \mathbf{b} \\
 & \theta = \max\{\mathbf{q}_i'(\mathbf{F}\mathbf{x} + \mathbf{g}) \,|\, \mathbf{q}_i \in Q\} \\
 & \mathbf{x} \geq \mathbf{0}
\end{array}$$

$$\begin{aligned} \text{minimize} \quad & \mathbf{c}'\mathbf{x}+\theta \\ \text{subject to} \quad & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \theta \geq \mathbf{q}_i'(\mathbf{F}\mathbf{x}+\mathbf{g}) \quad \forall \mathbf{q}_i \in Q \\ & 0 \geq \mathbf{w}_i'(\mathbf{F}\mathbf{x}+\mathbf{g}) \quad \forall \mathbf{w}_i \in W \\ & \mathbf{x} \geq \mathbf{0} \end{aligned} \tag{4.3}$$

Note that we needed to get rid of the "max" constraints, since they make the problem nonlinear. In model (4.3) though, since we have re-written the constraints, the problem is linear. Model (4.3) is what we choose to solve. Note that there are fewer variables than in model (4.1). Before there were n_2 $\mathbf{y}$ variables and now there is just one θ variable. Unfortunately, there are more constraints, since now we have a constraint for every extreme point and every extreme ray of the dual subproblem. Thus, we next use delayed constraint generation to solve model (4.3). We ignore the last set of constraints and solve the following model, called the restricted master problem.

$$\begin{aligned} \text{minimize} \quad & \mathbf{c}'\mathbf{x}+\theta \\ \text{subject to} \quad & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{aligned} \tag{4.4}$$

From solving this, we get an optimal solution $(\mathbf{x}^\star, \theta^\star)$. Now we need to check if $(\mathbf{x}^\star, \theta^\star)$ satisfies the ignored constraints. If it does, then $(\mathbf{x}^\star, \theta^\star)$ is the solution to model (4.3) with all the constraints included. Rather than checking each of the constraints one-by-one, which could be rather time-consuming, we can simply solve the dual subproblem

$$\begin{aligned} \text{maximize} \quad & \mathbf{p}'(\mathbf{F}\mathbf{x}+\mathbf{g}) \\ \text{subject to} \quad & \mathbf{p}'\mathbf{E} \leq \mathbf{f}' \end{aligned} \tag{4.5}$$

We get either a solution $\mathbf{p}^\star \in Q$ or we get an extreme ray $\mathbf{w}^\star \in W$. If the value at the solution is greater than θ, so $\theta < (\mathbf{p}^\star)'(\mathbf{F}\mathbf{x}^\star + \mathbf{g})$, then the constraint $\theta \geq (\mathbf{p})'(\mathbf{F}\mathbf{x} + \mathbf{g})$ in the model (4.3) is violated. This tells us that while ignoring that constraint, the optimal $\mathbf{x}$ value was not chosen. Thus we add the violated constraint to the restricted master problem (4.4). If we get an extreme point $\mathbf{w}^\star$, which means that the subproblem is infeasible, then $(\mathbf{w}^\star)'(\mathbf{F}\mathbf{x}^\star + \mathbf{g}) > \mathbf{0}$. We can add the violated constraint to the relaxed master problem, which becomes one of the following.

$$\begin{array}{ll} \text{minimize} & \mathbf{c}'\mathbf{x} + \theta \\ \text{subject to} & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \theta \geq (\mathbf{p}^\star)'(\mathbf{F}\mathbf{x} + \mathbf{g}) \\ & \mathbf{x} \geq \mathbf{0} \end{array}$$

$$\begin{array}{ll} \text{minimize} & \mathbf{c}'\mathbf{x} + \theta \\ \text{subject to} & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & 0 \geq (\mathbf{w}^\star)'(\mathbf{F}\mathbf{x}^\star + \mathbf{g}) \\ & \mathbf{x} \geq \mathbf{0} \end{array}$$

Now we re-solve this model, check for ignored constraints, and continue this process until no violated constraints are found. Then we have an optimal solution $(\mathbf{x}^\star, \mathbf{p}^\star)$ to (4.3). We can use complementary slackness to find $\mathbf{y}^\star$ from $\mathbf{p}^\star$. Then $(\mathbf{x}^\star, \mathbf{y}^\star)$ is the optimal solution to the original problem (4.1).

Note that there are two main reasons that we have used the dual for the second-stage problem both in the re-written model (4.3) and while checking for ignored constraints. One is that we get $\geq$ constraints in (4.3) instead of the $\leq$ constraints we would have if we used the primal subproblem. If we had $\leq$ constraints, the optimal value of (4.3) would be $-\infty$ since the objective function is being minimized. Second, in the dual subproblem, $\mathbf{x}$ is in the objective function instead of in the right hand side of the constraints as it is in the primal subproblem, so the set of extreme points of the polyhedron does not change as the $\mathbf{x}$ values change.

As mentioned above, Benders decomposition can be especially useful for two-stage stochastic problems. Then there is a set Ω of different outcomes that can occur after the first-stage decisions are made. In each of these outcomes, we have to solve a separate second-stage problem. Then, in using Benders decomposition, there are $|\Omega|$ different subproblems, and each contributes both optimality and feasibility constraints to the master problem. An original two-stage problem is below, where ρ^ω is the probability of outcome $\omega \in \Omega$ occurring. We include the second-stage problem for each different outcome, where $|\Omega| = K$.

$$\begin{aligned}
\text{minimize} \quad & \mathbf{c}'\mathbf{x} + \rho^1\mathbf{d}'\mathbf{y}^1 + \rho^2\mathbf{d}'\mathbf{y}^2 + ... + \rho^K\mathbf{d}'\mathbf{y}^K \\
\text{subject to} \quad & \mathbf{A}\mathbf{x} = \mathbf{b} \\
& \mathbf{E}\mathbf{y}^1 - \mathbf{F}^1\mathbf{x} = \mathbf{g}^1 \\
& \mathbf{E}\mathbf{y}^2 - \mathbf{F}^2\mathbf{x} = \mathbf{g}^2 \\
& \vdots \\
& \mathbf{E}\mathbf{y}^K - \mathbf{F}^K\mathbf{x} = \mathbf{g}^K \\
& \mathbf{x}, \mathbf{y}^1, \mathbf{y}^2, ..., \mathbf{y}^K \geq \mathbf{0}
\end{aligned}$$

Then the master problem becomes

$$\begin{aligned}
\text{minimize} \quad & \mathbf{c}'\mathbf{x} + \sum_{\omega\in\Omega} \rho^\omega\theta^\omega \\
\text{subject to} \quad & \mathbf{A}\mathbf{x}{=}\mathbf{b} \\
& \theta^\omega \geq (\mathbf{q}_i^\omega)'(\mathbf{F}^\omega\mathbf{x} + \mathbf{g}^\omega) && \forall\omega \in \Omega,\ \mathbf{q}_i^\omega \in Q^\omega \\
& 0 \geq (\mathbf{w}_i^\omega)'(\mathbf{F}^\omega\mathbf{x} + \mathbf{g}^\omega) && \forall\omega \in \Omega,\ \mathbf{w}_i^\omega \in W^\omega \\
& \mathbf{x} \geq \mathbf{0}
\end{aligned}$$

An explanation of Benders decomposition is in the book *Bertsimas and Tsitsiklis* (1997). Next, figure 4.1 illustrates the algorithm used for Benders decomposition.

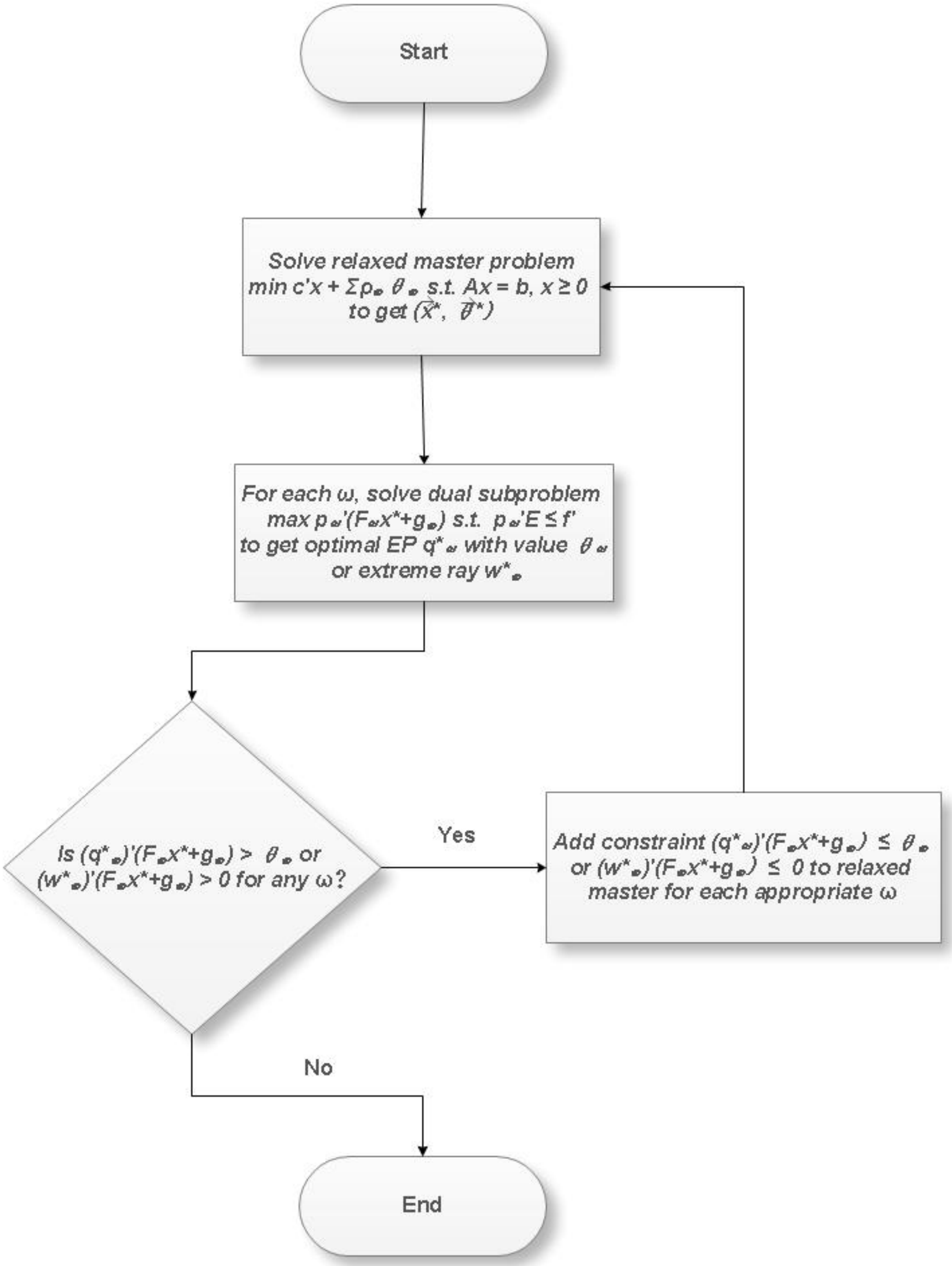
Start
Solve relaxed master problem
$\min c'x + \Sigma p_\omega \theta_\omega$ s.t. $Ax = b$, $x \geq 0$
to get $(\vec{x}^*, \vec{\theta}^*)$
For each ω, solve dual subproblem
$\max p_\omega'(F_\omega x^* + g_\omega)$ s.t. $p_\omega' E \leq f'$
to get optimal EP q^*_ω with value θ_ω
or extreme ray w^*_ω
Is $(q^*_\omega)'(F_\omega x^* + g_\omega) > \theta_\omega$ or
$(w^*_\omega)'(F_\omega x^* + g_\omega) > 0$ for any ω?
Yes
Add constraint $(q^*_\omega)'(F_\omega x^* + g_\omega) \leq \theta_\omega$
or $(w^*_\omega)'(F_\omega x^* + g_\omega) \leq 0$ to relaxed
master for each appropriate ω
No
End

Figure 4.1: Algorithm for Benders Decomposition

4.1.3 Passenger Reaccommodation

Now, we look at using Benders decomposition for our model in particular. The reason we use Benders decomposition is to decrease the solve time. For example, we want to be able to use flights from more than one day while taking into account the possible cost to the airline of getting hotel rooms and meals for passengers who are delayed overnight. Also, we want to be able to reroute many passengers and take into account many possibilities for the length of delay. Thus implementing Benders decomposition helps us be able to solve realistic problems. Recall that the model we are solving is

$$
\begin{aligned}
\text{minimize} \quad & \sum_{p\in P}\sum_{i\in I_p}\Big(1-\sum_{\omega\in\Omega}\epsilon_i^\omega\rho^\omega\Big)c_{pi}x_{pi}+\sum_{\omega\in\Omega}\rho^\omega\sum_{a\in A_e}c_a y_a^\omega && && (4.6)\\
\text{subject to} \quad & \sum_{i\in I_p}x_{pi}=1 && \forall p\in P\\
& \sum_{p\in P}\sum_{i\in I_p:f\in F_i}x_{pi}\le C_f && \forall f\in F\\
& \sum_{a\in B_1^\omega}y_a^\omega=\sum_{p\in P}\sum_{i\in I_p}\epsilon_i^\omega x_{pi} && \forall \omega\in\Omega\\
& \sum_{a\in B_n}y_a^\omega-\sum_{a\in E_n^\omega}y_a^\omega=0 && \forall n\in N\setminus(\{n_1\}\cup N_p),\ \omega\in\Omega\\
& \sum_{a\in E_{n_p}}y_a^\omega=\sum_{i\in I_p}\epsilon_i^\omega x_{pi} && \forall p\in P,\ \omega\in\Omega\\
& y_a^\omega\le u_a-\sum_{p\in P}\sum_{i\in I_p:a\in A_{fpi}}(1-\epsilon_i^\omega)x_{pi} && \forall a\in A_{fp},\ \omega\in\Omega\\
& x_{pi}\in\{0,1\} && \forall p\in P,\ \forall i\in I_p\\
& y_a^\omega\ge 0 && \forall \omega\in\Omega,\ a\in A^\omega
\end{aligned}
$$

The first-stage and second-stage problem for each $\omega\in\Omega$ are

$$\begin{array}{lll}
\text{minimize} & \displaystyle\sum_{p\in P}\sum_{i\in I_p}\Big(1-\sum_{\omega\in\Omega}\epsilon_i^{\omega}\rho^{\omega}\Big)c_{pi}x_{pi} & \\
\text{subject to} & \displaystyle\sum_{i\in I_p} x_{pi}=1 & \forall p\in P \\
& \displaystyle\sum_{p\in P}\sum_{i\in I_p: f\in F_i} x_{pi}\le C_f & \forall f\in F \\
& x_{pi}\in\{0,1\} & \forall p\in P,\ \forall i\in I_p
\end{array}$$

$$\begin{array}{lll}
\text{minimize} & \displaystyle\sum_{a\in A_e} c_a y_a^{\omega} & \\
\text{subject to} & \displaystyle\sum_{a\in B_1^{\omega}} y_a^{\omega}=\sum_{p\in P}\sum_{i\in I_p}\epsilon_i^{\omega}x_{pi} & \\
& \displaystyle\sum_{a\in B_n} y_a^{\omega}-\sum_{a\in E_n^{\omega}} y_a^{\omega}=0 & \forall n\in N\setminus(\{n_1\}\cup N_p) \\
& \displaystyle\sum_{a\in E_{n_p}} y_a^{\omega}=\sum_{i\in I_p}\epsilon_i^{\omega}x_{pi} & \forall p\in P \\
& \displaystyle y_a^{\omega}\le u_a-\sum_{p\in P}\sum_{i\in I_p: a\in A_{fpi}}(1-\epsilon_i^{\omega})x_{pi} & \forall a\in A_{fp} \\
& y_a^{\omega}\ge 0 & \forall a\in A^{\omega}
\end{array}$$

To form the dual of the second-stage problem (the subproblem), we need the following notation for each $\omega\in\Omega$:

$$\delta_a^{\omega}=\begin{cases}1 & \text{arc}\,a\,\text{begins at the start node in outcome}\,\omega\\ 0 & \text{otherwise}\end{cases}$$

$$\zeta_{n,a}^{\omega}=\begin{cases}1 & \text{arc}\,a\,\text{begins at flight node}\,n\,\text{in outcome}\,\omega\\ 0 & \text{otherwise}\end{cases}$$

$$\lambda_{n,a}^{\omega}=\begin{cases}1 & \text{arc}\,a\,\text{ends at flight node}\,n\,\text{in outcome}\,\omega\\ 0 & \text{otherwise}\end{cases}$$

$$\sigma_{p,a}^{\omega}=\begin{cases}1 & \text{arc}\,a\,\text{ends at end node for passenger}\,p\,\text{in outcome}\,\omega\\ 0 & \text{otherwise}\end{cases}$$

$$\eta_{i,f} = \begin{cases} 1 & \text{flight } f \text{ is part of itinerary } i \\ 0 & \text{otherwise} \end{cases}$$

$$\kappa_{f,a} = \begin{cases} 1 & \text{arc } a \text{ is between the two nodes for flight } f \\ 0 & \text{otherwise} \end{cases}$$

Last, we let $c_a = 0$ for $a \notin A_e$. Using this new notation and suppressing the ω for simplicity, our second-stage problem for each $\omega \in \Omega$ can be written

$$\begin{array}{lll}
\text{minimize} & \displaystyle\sum_{a \in A} c_a y_a & \\
\text{subject to} & \displaystyle\sum_{a \in A} \delta_a \cdot y_a - \sum_{p \in P} \sum_{i \in I_p} \epsilon_i x_{pi} = 0 & \\
& \displaystyle\sum_{a \in A} (\zeta_{n,a} - \lambda_{n,a}) \cdot y_a = 0 & \forall n \in N \setminus (\{n_1\} \cup N_p) \\
& \displaystyle\sum_{a \in A} \sigma_{p,a} \cdot y_a - \sum_{i \in I_p} \epsilon_i x_{pi} = 0 & \forall p \in P \\
& \displaystyle\sum_{a \in A} \kappa_{f,a} \cdot y_a + \sum_{p \in P} \sum_{i \in I_p} \eta_{i,f} (1 - \epsilon_i) x_{pi} \ \le C_f & \forall f \in F \\
& y_a \ge 0 & \forall a \in A^\omega
\end{array}$$

Our subproblem for each $\omega \in \Omega$ is in the form

$$\begin{array}{ll}
\text{minimize} & \mathbf{d}'\mathbf{y} \\
\text{subject to} & \mathbf{E}\mathbf{y} - \mathbf{F}\mathbf{x} \ = \ \mathbf{g} \\
& \mathbf{H}\mathbf{y} - \mathbf{J}\mathbf{x} \ \le \ \mathbf{k} \\
& \mathbf{y} \ \ge \mathbf{0}
\end{array}$$

The dual of this is

$$\begin{aligned}
\text{maximize} \quad & \begin{bmatrix} \boldsymbol{\pi} & \boldsymbol{\mu} \end{bmatrix}' \begin{bmatrix} \mathbf{g} & \mathbf{Fx} \\ \mathbf{k} & \mathbf{Jx} \end{bmatrix} \\
\text{subject to} \quad & \begin{bmatrix} \boldsymbol{\pi} & \boldsymbol{\mu} \end{bmatrix}' \begin{bmatrix} \mathbf{E} \\ \mathbf{H} \end{bmatrix} \leq \mathbf{d}' \\
& \boldsymbol{\mu} \leq \mathbf{0}
\end{aligned}$$

Note that the variables are α, $\boldsymbol{\beta}$, $\boldsymbol{\pi}$, and $\boldsymbol{\mu}$. Also, the length of each of these is 1, $2*|F|$, $|P|$, and $|F|$, respectively. Thus the dual of our second-stage problem for each $\omega \in \Omega$ is

$$\begin{aligned}
\text{maximize} \quad & \left(\sum_{p \in P} \sum_{i \in I_p} \epsilon_i x_{pi}\right) \cdot \alpha + \sum_{p=1}^{|P|} \left(\sum_{i \in I_p} \epsilon_i x_{pi}\right) \cdot \pi_p + \\
& \sum_{f=1}^{|F|} \left(C_f - \sum_{p \in P} \sum_{i \in I_p} \eta_{i,f}(1-\epsilon_i) x_{pi}\right) \cdot \mu_f \\
\text{subject to} \quad & \delta_a \cdot \alpha + \sum_{n=1}^{2*|F|} (\zeta_{n,a} - \lambda_{n,a}) \cdot \beta_n + \sum_{p=1}^{|P|} \sigma_{p,a} \cdot \pi_p + \sum_{f=1}^{|F|} \kappa_{f,a} \cdot \mu_f \leq c_a, \quad \forall a \in A \\
& \boldsymbol{\mu} \leq 0
\end{aligned} \tag{4.7}$$

Now that we have our dual subproblem, the formulation (4.3), the master problem, becomes

$$
\begin{aligned}
\text{minimize} \quad & \sum_{p\in P}\sum_{i\in I_p}\Big(1-\sum_{\omega\in\Omega}\epsilon_i^{\omega}\rho^{\omega}\Big)c_{pi}x_{pi}+\sum_{\omega\in\Omega}\rho^{\omega}\theta^{\omega} && && (4.8)\\
\text{subject to} \quad & \sum_{i\in I_p}x_{pi}=1 && \forall p\in P\\
& \sum_{p\in P}\ \sum_{i\in I_p:f\in F_i}x_{pi}\le C_f && \forall f\in F\\
& \theta^{\omega}\ge\Big(\sum_{p\in P}\sum_{i\in I_p}\epsilon_i^{\omega}x_{pi}\Big)\cdot\psi_j^{\omega}+\sum_{p=1}^{|P|}\Big(\sum_{i\in I_p}\epsilon_i^{\omega}x_{pi}\Big)\cdot\tau_{jp}^{\omega}+\\
& \qquad \sum_{f=1}^{|F|}\Big(C_f-\sum_{p\in P}\sum_{i\in I_p}\eta_{i,f}(1-\epsilon_i^{\omega})x_{pi}\Big)\cdot\nu_{jf}^{\omega}\\
& \qquad \forall\ \big[\psi_j^{\omega}\ \ \boldsymbol{\phi}_j^{\omega}\ \ \boldsymbol{\tau}_j^{\omega}\ \ \boldsymbol{\nu}_j^{\omega}\big]\in Q^{\omega},\ \omega\in\Omega
\end{aligned}
$$

where Q^{ω} for all $\omega\in\Omega$ is the set of extreme points of the polyhedron

$$
\Big\{\big[\alpha^{\omega}\ \boldsymbol{\beta}^{\omega}\ \boldsymbol{\pi}^{\omega}\ \boldsymbol{\mu}^{\omega}\big]\ \mid\ \delta_a\cdot\alpha+\sum_{n=1}^{2*|F|}(\zeta_{n,a}-\lambda_{n,a})\cdot\beta_n+\sum_{p=1}^{|P|}\sigma_{p,a}\cdot\pi_p+\sum_{f=1}^{|F|}\kappa_{f,a}\cdot\mu_f\ \le\ c_a
$$
$$
\forall a\in A^{\omega},\ \boldsymbol{\mu}^{\omega}\ \le\ \mathbf{0}\Big\}
$$

Note that there are no feasibility constraints involving the extreme rays in this master problem. This is because we add an arc from each start node to each end passenger node. Flow on one of these arcs represents the passenger not arriving at the desired location. We assign a very high cost to these arcs, so that they are not chosen unless it is not possible to get all passengers to their destinations. Needing to use these arcs in the second-stage problem is not optimal, and using them results in optimality constraints being added to the restricted master problem.

Also note that while solving the mixed-integer master problem, the method of branch and bound may be used because of the binary first-stage variables. Note that this is an exact algorithm and is guaranteed to produce an optimal solution even if it takes an exponential number of iterations, as mentioned on page 480 of the book *Bertsimas and Tsitsiklis* (1997).

Next, as described above, we use delayed constraint generation to solve this model. We ignore the last set of constraints and solve the restricted master problem:

$$\begin{aligned}
\text{minimize} \quad & \sum_{p\in P}\sum_{i\in I_p}\Big(1-\sum_{\omega\in\Omega}\epsilon_i^\omega\rho^\omega\Big)c_{pi}x_{pi}+\sum_{\omega\in\Omega}\rho^\omega\theta^\omega && && (4.9)\\
\text{subject to} \quad & \sum_{i\in I_p}x_{pi}=1 && \forall p\in P\\
& \sum_{p\in P}\sum_{i\in I_p:f\in F_i}x_{pi}\le C_f && \forall f\in F
\end{aligned}$$

From solving this, we get an optimal solution $(\mathbf{x}^\star,\boldsymbol{\theta}^\star)$. Next we check if $(\mathbf{x}^\star,\boldsymbol{\theta}^\star)$ satisfies the ignored constraints. If it does, then $(\mathbf{x}^\star,\boldsymbol{\theta}^\star)$ is the solution to model (4.8) with all the constraints included. Rather than checking each of the constraints one-by-one, which could be rather time-consuming, we can simply solve the dual subproblem (4.7) for each $\omega\in\Omega$. From solving model (4.7), for each $\omega\in\Omega$ we get a solution $[\alpha^\star\ \boldsymbol{\beta}^{\omega\star}\ \boldsymbol{\pi}^{\omega\star}\ \boldsymbol{\mu}^{\omega\star}]$, which is one of the extreme points of Q^ω, with value

$$\Big(\sum_{p\in P}\sum_{i\in I_p}\epsilon_i^\omega x_{pi}\Big)\alpha^{\omega\star}+\sum_{p=1}^{|P|}\Big(\sum_{i\in I_p}\epsilon_i^\omega x_{pi}\Big)\pi_p^{\omega\star}+\sum_{f=1}^{|F|}\Big(C_f-\sum_{p\in P}\sum_{i\in I_p}\eta_{i,f}(1-\epsilon_i^\omega)x_{pi}\Big)\mu_f^{\omega\star}$$

If this value is greater than θ, then the constraint

$$\theta^\omega\ge\Big(\sum_{p\in P}\sum_{i\in I_p}\epsilon_i^\omega x_{pi}\Big)\psi_j^\omega+\sum_{p=1}^{|P|}\Big(\sum_{i\in I_p}\epsilon_i^\omega x_{pi}\Big)\tau_{jp}^\omega+\sum_{f=1}^{|F|}\Big(C_f-\sum_{p\in P}\sum_{i\in I_p}\eta_{i,f}(1-\epsilon_i^\omega)x_{pi}\Big)\nu_{jf}^\omega$$

in model (4.8) is violated for $[\alpha^{\omega\star}\ \boldsymbol{\beta}^{\omega\star}\ \boldsymbol{\pi}^{\omega\star}\ \boldsymbol{\mu}^{\omega\star}]=[\psi_j^\omega\ \boldsymbol{\phi}_j^\omega\ \boldsymbol{\tau}_j^\omega\ \boldsymbol{\nu}_j^\omega]$ $(\theta\ge(\mathbf{p})'(\mathbf{F}\mathbf{x}+\mathbf{g})$ in model (4.3)). This tells us that while ignoring that constraint, the optimal $\mathbf{x}$ value was not chosen. Thus we add the violated constraints to the restricted master problem (4.9). If $\Omega_1\in\Omega$ is the set of outcomes with violated optimality constraints, then the restricted master problem becomes

$$
\begin{aligned}
\text{minimize} \quad & \sum_{p\in P}\sum_{i\in I_p}\Big(1-\sum_{\omega\in\Omega}\epsilon_i^{\omega}\rho^{\omega}\Big)c_{pi}x_{pi}+\sum_{\omega\in\Omega}\rho^{\omega}\theta^{\omega} \\
\text{subject to} \quad & \sum_{i\in I_p}x_{pi}=1 && \forall p\in P \\
& \sum_{p\in P}\sum_{i\in I_p:f\in F_i}x_{pi}\leq C_f && \forall f\in F \\
& \theta^{\omega}\geq\Big(\sum_{p\in P}\sum_{i\in I_p}\epsilon_i^{\omega}x_{pi}\Big)\alpha^{\omega\star}+\sum_{p=1}^{|P|}\Big(\sum_{i\in I_p}\epsilon_i^{\omega}x_{pi}\Big)\pi_p^{\omega\star}+ \\
& \quad \sum_{f=1}^{|F|}\Big(C_f-\sum_{p\in P}\sum_{i\in I_p}\eta_{i,f}(1-\epsilon_i^{\omega})x_{pi}\Big)\mu_f^{\omega\star} && \forall\omega\in\Omega_1
\end{aligned}
$$

Now we re-solve this model, check for ignored constraints, and continue this process until no violated constraints are found. Then we have an optimal solution to (4.8). We can use complementary slackness to find $\mathbf{y}^{1\star},\mathbf{y}^{2\star},...,\mathbf{y}^{|\Omega|\star}$. Then $(\mathbf{x}^{\star},\mathbf{y}^{1\star},\mathbf{y}^{2\star},...,\mathbf{y}^{|\Omega|\star})$ is the optimal solution to the original problem (4.6).

Recall that our two-stage stochastic problem has multiple subproblems, one for each possibility for the length of delay. We check each subproblem for an ignored constraint, add all ignored ones to the restricted master problem, and re-solve. In following the algorithm for Benders decomposition, we could instead add any ignored optimality constraints from just one subproblem and re-solve the master problem, then repeat the same procedure for each of the other subproblems. We have found that the solve time is shorter using the first method, since solving the restricted master problem takes more time than solving each of the subproblems. Thus, one iteration represents checking all subproblems, adding ignored constraints, and solving the restricted master problem.

4.1.4 Analysis of Added Constraints

We next consider the question of what adding constraints to the restricted master problem rules out and what the constraints mean. In general, if a constraint is added, it means that the first-stage variables were not chosen in such a way that led to the best possible second-stage decisions. We need to chose different first-stage variables to achieve the best solution. The constraint rules out that particular combination of first-stage variables from being chosen again.

In general, there are two types of constraints that can be generated to add to the restricted master problem. Optimality constraints are added when the first-stage solution leads to a sub-optimal second-stage solution, and feasibility constraints are added when the first-stage solution makes the second-stage problem infeasible. Recall that our model is set up in such a way that the only constraints added are optimality constraints, and there are no feasibility constraints added. We forced this situation by adding an arc to each subproblem's network from the start node to each end node, and flow on one of these arcs has a high cost and represents the corresponding passenger not being reaccommodated. Then there is always a feasible flow over the network. We did this because optimality constraints are easier to code and interpret. Suppose in a second-stage solution one or more of these arcs is used. An optimality constraint is then added, since using these high-cost arcs is not optimal unless the whole two-stage problem really has no solution with actually itineraries to give to the passengers in that scenario.

For example, for instance five described in section 4.2.3, we ran several simulations using the large data set describe in section 4.2.1. In the first iteration for one of these simulations, after solving the subproblem for $\omega = 11$, the following constraint was added to the restricted master problem.

$$\theta^{11} \geq 4320(x_{3,1} + x_{3,4} + x_{3,5}) + 208(x_{6,1} + x_{6,2}) + 4320(x_{7,0} + x_{7,2} + x_{7,3}) \tag{4.10}$$

By looking at the solution file, we can see that the only x variables in this inequality that were positive before this constraint was generated were $x_{6,2}$ and $x_{7,0}$, corresponding to passengers 6 and 7. Recall that in general x_{pi} corresponds to passenger $p \in P$

and the i'th itinerary in p's list of possible first-stage itineraries. Note that this inequality deals with scenario $\omega = 11$, since the left side has the variable θ^{11}. We next follow the experiences of passengers 6 and 7 before and after this constraint was added.

- For passenger 7, we see that the flights corresponding to $x_{7,0}$ caused a misconnection in outcome $\omega = 11$, since the cost for the 0'th first-stage itinerary is 4320 (three days) in the equation, which is the cost of non-arrival. Thus, an arc representing non-arrival had to be used for passenger 7. After the constraint was added and the restricted master problem was re-solved, $x_{7,1}$ was chosen instead with a cost of 57 minutes. This was a big improvement and moved the passenger off the delayed flight in the first stage. Although this cannot be seen by the equation, $n_{11,7}$ became 0, showing that the non-arrival arc was not used.

- For passenger 6, choosing itinerary 2 by letting $x_{6,2}$ be 1 before adding the constraint led to a cost of 208 minutes in the second-stage problem. We can see this since 208 is the cost of $x_{6,2}$ in the constraint. The value of $x_{6,2}$ did not actually change at the last iteration, so this was the best choice considering all passengers' options.

Recall that the part of the objective function for a first-stage itinerary $i \in I_p$ for a passenger $p \in P$ is $\left(1 - \sum_{\omega \in \Omega} \epsilon_i^\omega \rho^\omega\right) c_{pi}$. Suppose that the passenger is disrupted with this itinerary in all scenarios except $\omega = 0$, so $\epsilon_i^\omega = 1$ for all $\omega \neq 0$. Then the cost becomes $\rho^0 c_{pi}$, which may be very small. Unfortunately, then itinerary i may be chosen the first time the restricted master problem is solved, even though it is a bad choice. There will be a high cost incurred in the second stage for choosing this itinerary, and then an optimality constraint will be generated to preclude itinerary i from being chosen. This type of situation is shown in the example above. The first time the restricted master problem was solved, passenger 7 received itinerary 0, which caused a misconnection in several scenarios. An optimality constraint was generated, and a better choice was made. Even though the solution to the original restricted master problem may not make sense intuitively, the correct solution will be chosen.

Suppose that the capacities on all flights are effectively infinite, so that there is enough capacity on all flights to accommodate any passengers that could take those flights. The capacity constraints can then be ignored, since there is no competition between passengers to get the best seats. In this case, we can actually break the whole model down into many separate models, one for each passenger, since there is no interaction between passengers. If we still use Benders decomposition on each problem, will we automatically choose the best first-stage decisions or will we have optimality constraints added to the restricted master problems? Since in the first stage, we pretend that the original delayed flight is not actually delayed, then most passengers will simply be assigned their original itineraries the first time the restricted master problem is solved. A description of cases when passengers are not assigned their original itineraries is in this chapter. Often this is not the best choice for passengers because the delay could be long and could cause a misconnection, so optimality constraints will still be added, even without capacity constraints.

4.1.5 Analysis of Iterations

In section 4.1.3, we presented the method of Benders decomposition for solving the problem in section 3.2.6 modeling the reaccommodation of disrupted airline passengers. We then explored the types of constraints generated and their meanings in section 4.1.4. Next, we consider limits on how long the process may take to solve the model, including the total number of constraints that may be generated.

In general, while carrying out Benders decomposition, we temporarily ignore the optimality and feasibility constraints for the subproblems, solve the restricted master problem, and check for an ignored constraint from each subproblem. We then add any that are violated, and re-solve the restricted master problem. In each iteration of Benders decomposition, only one constraint from each subproblem may be added. Recall that for the model from section 3.2.6, only optimality constraints may be generated, not feasibility constraints, as explained in sections 4.1.3 and 4.1.4. The ignored optimality constraints for the subproblem for each outcome $\omega \in \Omega$ are

$$\begin{aligned}\theta^{\omega} \geq\ & \left(\sum_{p\in P}\sum_{i\in I_p}\epsilon_i^{\omega}x_{pi}\right)\cdot\psi_j^{\omega} + \sum_{p=1}^{|P|}\left(\sum_{i\in I_p}\epsilon_i^{\omega}x_{pi}\right)\cdot\tau_{jp}^{\omega} + \\ & \sum_{f=1}^{|F|}\left(C_f - \sum_{p\in P}\sum_{i\in I_p}\eta_{i,f}(1-\epsilon_i^{\omega})x_{pi}\right)\cdot\nu_{jf}^{\omega}, \quad \forall\ \left[\psi_j^{\omega}\ \ \boldsymbol{\phi}_j^{\omega}\ \ \boldsymbol{\tau}_j^{\omega}\ \ \boldsymbol{\nu}_j^{\omega}\right] \in Q^{\omega}\end{aligned}$$

For each scenario $\omega \in \Omega$, Q^{ω} is the set of extreme points for the dual subproblem, so there is one optimality constraint for each extreme point. Thus the number of extreme points in a certain outcome is the number of constraints that may be generated from the subproblem. The number of extreme points give us a worst-case situation for how many iterations there may be. We know how many constraints are in each dual subproblem, shown as model (4.7) in section 4.1.3, and that number does not change throughout the process of Benders decomposition. We can thus find the number of extreme points in each dual subproblem, since they are all basic feasible solutions.

The number of extreme points for each dual subproblem depends on the number of constraints, and there is one constraint for every arc in the network used for the corresponding primal subproblem. The details of the data set of flights determine the number of arcs, which can be calculated. For the minimum cost flow model for the deterministic problem from section 3.1.3, we found in section 3.1.4 that the number of variables, which is the number of arcs, is $O(F^3 + FP)$. The second-stage dual subproblems thus have this many constraints, since each primal subproblem is the same type of model as the one in section 3.1.3. The chart in section 3.1.4 shows the number of variables for the minimum cost flow problem with different data sets.

We see that the number of possible iterations for Benders decomposition can be quite large. In practice, though, the number of iterations is much smaller than expected. In section 4.3, we explore how many iterations each simulation takes in different instances. We see that there is normally less than 8 iterations. The situations when there are more or less is affected by the available capacities on the flights and are discussed in section 4.3.

4.1.6 Example

We will solve the problem below with Benders decomposition by hand as done in the notes from IOE 591 (*Cohn* (2009)). For simplicity, there is only one possible outcome in the set Ω.

$$\begin{aligned} \text{minimize} \quad & x_1 + 2x_2 + 3x_3 + y_1 + 5y_2 + 10y_3 \\ \text{subject to} \quad & x_1 + x_2 + x_3 = 1 \\ & x_1 + y_1 \leq 1 \\ & x_2 + y_2 \leq 1 \\ & x_3 + y_3 \leq 1 \\ & y_1 + y_2 + y_3 = 1 \\ & x_1, x_2, x_3, y_1, y_2, y_3 \geq 0 \end{aligned}$$

Note that we can tell that the optimal solution is $(0, 1, 0, 1, 0, 0)$ with value 3. The first stage problem in x is

$$\begin{aligned} \text{minimize} \quad & x_1 + 2x_2 + 3x_3 \\ \text{subject to} \quad & x_1 + x_2 + x_3 = 1 \\ & x_1, x_2, x_3 \geq 0 \end{aligned}$$

The second stage problem, called the subproblem, which has optimal value z^P, is

$$\begin{aligned} \text{minimize} \quad & y_1+5y_2+10y_3 \\ \text{subject to} \quad & y_1 \leq 1 - x_1 \\ & y_2 \leq 1 - x_2 \\ & y_3 \leq 1 - x_3 \\ & y_1 + y_2 + y_3 = 1 \\ & y_1, y_2, y_3 \geq 0 \end{aligned} \tag{4.11}$$

The dual of this subproblem, with optimal value $z^D = z^P$, is

$$\begin{aligned} \text{maximize} \quad & (1-x_1)p_1 + (1-x_2)p_2 + (1-x_3)p_3 + p_4 \\ \text{subject to} \quad & p_1 + p_4 \leq 1 \\ & p_2 + p_4 \leq 5 \\ & p_3 + p_4 \leq 10 \\ & p_1, p_2, p_3 \leq 0 \\ & p_4 \text{ free} \end{aligned}$$

As described above, we form the problem

$$\begin{aligned} \text{minimize} \quad & x_1 + 2x_2 + 3x_3 + z \\ \text{subject to} \quad & x_1 + x_2 + x_3 = 1 \\ & (1-x_1)p_1^i + (1-x_2)p_2^i + (1-x_3)p_3^i + p_4^i \leq z \qquad \forall i \in A \\ & (1-x_1)w_1^j + (1-x_2)w_2^j + (1-x_3)w_3^j + w_4^j \leq 0 \qquad \forall j \in B \\ & x_1, x_2, x_3 \geq 0 \end{aligned}$$

We ignore the second and third sets of constraints, so we want to solve

$$\begin{aligned} \text{minimize} \quad & x_1 + 2x_2 + 3x_3 + z \\ \text{subject to} \quad & x_1 + x_2 + x_3 = 1 \\ & x_1, x_2, x_3 \geq 0 \end{aligned} \tag{4.12}$$

We can tell that the optimal solution is $(1, 0, 0, -\infty)$ with value $-\infty$. Next, we check if any of the ignored constraints are violated. If it is true that $z \geq z^D(1, 0, 0)$, then we have found the optimal solution. We plug $(x_1, x_2, x_3) = (1, 0, 0)$ into the primal subproblem (4.11) above and get

$$\begin{aligned} \text{minimize} \quad & y_1 + 5y_2 + 10y_3 \\ \text{subject to} \quad & y_1 \leq 0 \\ & y_2 \leq 1 \\ & y_3 \leq 1 \\ & y_1 + y_2 + y_3 = 1 \\ & y_1, y_2, y_3 \geq 0 \end{aligned}$$

The optimal solution is $(0, 1, 0)$ with value 5 by inspection. Also, the dual of the above problem, is

$$\begin{aligned} \text{maximize} \quad & p_2 + p_3 + p_4 \\ \text{subject to} \quad & p_1 + p_4 \leq 1 \\ & p_2 + p_4 \leq 5 \\ & p_3 + p_4 \leq 10 \\ & p_1, p_2, p_3 \leq 0 \\ & p_4 \text{ free} \end{aligned}$$

Using complementary slackness or inspection, the optimal solution is $(-4, 0, 0, 5)$ with value 5. Since $5 > -\infty$, we must add to the restricted master problem (4.12) the constraint $z \geq (1 - x_1)(-4) + (1 - x_2)(0) + (1 - x_3)(0) + 5 = 4x_1 + 1$. Thus the new restricted master problem is

$$\begin{array}{ll}
\text{minimize} & x_1 + 2x_2 + 3x_3 + z \\
\text{subject to} & x_1 + x_2 + x_3 = 1 \\
& z \geq 4x_1 + 1 \\
& x_1, x_2, x_3 \geq 0
\end{array}$$

The optimal solution is $(0, 1, 0, 1)$ with value 3. Now we start the whole process again. The subproblem (4.11) now is

$$\begin{array}{ll}
\text{minimize} & y_1 + 5y_2 + 10y_3 \\
\text{subject to} & y_1 \leq 1 \\
& y_2 \leq 0 \\
& y_3 \leq 1 \\
& y_1 + y_2 + y_3 = 1 \\
& y_1, y_2, y_3 \geq 0
\end{array}$$

The optimal solution is $(1, 0, 0)$ and has value 1. The dual of this problem is

$$\begin{array}{ll}
\text{maximize} & p_1 + p_3 + p_4 \\
\text{subject to} & p_1 + p_4 \leq 1 \\
& p_2 + p_4 \leq 5 \\
& p_3 + p_4 \leq 10 \\
& p_1, p_2, p_3 \leq 0 \\
& p_4 \text{ free}
\end{array}$$

The optimal solution is $(0, 0, 0, 1)$ with value 1. Since $z = 1$, we have arrived at the optimal solution, which is $(x_1, x_2, x_3, y_1, y_2, y_3) = (0, 1, 0, 1, 0, 0)$. Note that this is the same solution that was found at the beginning of this section.

4.1.7 Literature Review

We have presented the method of Benders decomposition, discussed using it for our model for passenger reaccommodation, and looked at a small example. We now discuss four situations where Benders decomposition has been used in the past to help solve an airline planning or recovery problem.

First, Benders decomposition is used to solve the integrated aircraft routing problem and crew scheduling problem in the paper *Cordeau et al.* (2001). The aircraft routing problem is the first stage problem and the crew scheduling problem is the subproblem. In the paper *Mercier et al.* (2005), the authors solve the same integrated problem. They use Benders decomposition to solve the model in two different ways, where each of the aircraft routing problem and crew pairing problem is the master problem in one of the two approaches. They also present strategies to improve the generation of cuts in order to lower the number of iterations that must be completed.

As mentioned in chapter II, in the paper *Petersen et al.* (2010), the authors use Benders decomposition to solve an integrated recovery problem involving the schedule, aircraft, crew, and passenger recovery problems. The restricted master problems is the schedule generation problem, and the aircraft recovery model, the crew recovery model, and the passenger recovery model are the subproblems. Note that the three subproblems are independent of each other since they each depend only on the flight schedule. In the master problem, the variables represent strings of flights operated by a single aircraft at certain times, and the decision variables are binary. The authors have five types of Benders cuts, which are feasibility cuts for all three subproblems and optimality cuts for the crew and passenger recovery problems. Since the authors use column generation along with Benders decomposition, sometimes their cuts become infeasible, so they sometimes remove the Benders cuts when new columns are added. The authors find that their integrated model is effective when no more than 65% of flights could be delayed and the time is limited to one day. Also, the results are better than those from a sequential model.

In the future research section of the paper *Rosenberger et al.* (2003), the authors discuss using a two-stage stochastic programming problem for aircraft recovery, where the stochasticity comes from weather possibilities. The authors plan to use Benders decomposition to solve this model to lessen the run time.

4.2 Computational Results

In this section, we present and analyze several facets of our computational results. First, we discuss the data used for our implementation and give some details about our simulations. Next, we want to compare our method with a standard method used for reaccommodation in practice. We simulate this method and then compare the delays passengers experience from using both methods. The perspective of an airline and the value of our solutions are discussed. Last, we also analyze results concerning the run time in different situations, including comparing solving the model as a mixed-integer model all at once and using Benders decomposition.

4.2.1 Data

For most of our instances described in section 4.2.3, we downloaded the flight data for Delta Airlines on January 6, 2010 from the Bureau of Transportation Statistics (*http://www.transtats.bts.gov*). This data includes most of the information to input in the models, such as the origin city, destination city, departure time, and flight time for all 1144 flights. The flight we use for the delayed flight is numbered 33, flew from John F. Kennedy International Airport (JFK) to Hartsfield-Jackson Atlanta International Airport (ATL), departed at 8:20am, and arrived at 11:08am.

Information about passengers' planned itineraries is not readily available, so we developed some data sets. For different instances, we use a certain number of different locations for the passengers' final destinations, usually 15 or 30, then distribute the passengers over these locations. A more thorough description of the passengers' final destinations is given in section 4.2.3.

Also, we have to choose the passengers' original itineraries that have connecting flights. For each simulation, we give each passenger the best or second best itinerary going to his or her final destination. The best itinerary is simply the one that will arrive earliest, not taking into account any delays, since those are not known when passengers buy their tickets. Note that the amount of time saved by solving the model may actually be greater in practice than shown in the simulations in section 4.2.3, since most likely not all passengers begin with the best or second best itinerary to their destinations due to cost or capacities.

We also have to decide on the number of available seats on the non-delayed flights. In each instance, we allow the same available capacity for all the flights, and it ranges from 1 to 12. Although in practice flights do not have the same available capacities, this assumption allows us to easily evaluate the effect of changing available capacities on the results and run times. Also, many times when flights have higher available capacities, the model is not affected, based on the destinations of the passengers on the delayed flight. Note that according to the BTS website (*http://www.transtats.bts.gov*), the total load factor, defined as passenger-miles as a proportion of available seat-miles, was 83.03% for domestic flights in 2011. In particular, Southwest, American, US, and Delta Airlines had average load factors ranging from 80.82% to 86.69% in 2011. Considering the size of most aircraft used, the available capacities we have for our simulations are probably conservative, and they are most likely normally higher in practice.

Last, while implementing the model, we created a small test data set of only eight flights and three passengers, which can be solved by hand, since the model has few variables and constraints. It also helps us understand the results more clearly.

4.2.2 Simulation Approaches

Recall that in our approach to reaccommodating passengers, one advantage is that we preemptively reroute passengers, so that passengers can get off the delayed flight even before the delay length is known. Second, we consider all passengers together, so that if one passenger has only one option to reach a connection in time, that person can be given priority. We contrast this with a method where passengers cannot get off the delayed flight. Also, after it has landed, any passengers that have missed connections are reaccommodated one-by-one, not necessarily in any particular order. We call this method the "current" method, and our method the "new" method.

To make this comparison, we simulate the current approach. We have a one-stage problem with the same possibilities and probabilities for the length of delay as in our model, but we make all passengers keep the original flight, then reaccommodate them one-by-one after they land if necessary. In our simulations, the order that we reaccommodate the passengers in is randomly generated, and for each person, we choose the itinerary that will get him or her to the destination earliest using flights with available seats remaining.

In each outcome for the length of delay, we compare what time all the passengers arrive and how much they were delayed using the current method and using the new method. We calculate the average delays experienced by the passengers on the delayed flight using both methods. We control different variables to see the differences in the outcomes for the solutions. Variables that we change are the number of flights in the data set, the number of passengers on the delayed flight, the lengths of possible delays, the number of possible scenarios, the available capacities on the flights that are not delayed, the final destinations of the passengers, and the order that the passengers are reaccommodated in the current method. One interesting way to evaluate the effectiveness of our method is to see how many passengers are moved from the delayed flight in the first-stage problem.

Before we look at the results, it is helpful to keep in mind the three ways that passengers can be delayed and thus cost is contributed to the objective value.

- A passenger whose original itinerary consists of only the delayed flight stays on it in the first-stage problem. The passenger's cost is equal to the delay that actually occurs.

- If a passenger originally has more than one flight in his or her itinerary, then in the first-stage problem, he or she stays on the delayed flight and then misses the connection. The passenger is then given a new second-stage itinerary that arrives later than the passenger had originally planned to arrive. This length of time is the cost for the passenger.

- A passenger is moved off the delayed flight in the first-stage problem and given another itinerary that arrives a certain length of time later than the passenger planned to arrive. This length of time is the passenger's cost.

Using the new method, all three of these ways are possible. In the current method, there is delay only from the first two ways, since we assume no passengers are moved off the delayed flight.

4.2.3 Results

Tables 4.1 and 4.2 below summarize the different instances we ran for our computational results. The first row, "Num Pass," gives the number of passengers on the delayed flight, and the second row, "Num Fts," is the number of flights in the data set used for the instance. The number of final destinations is how many different destinations the passengers on the delayed flight are headed to, which is shown in the third row. The row "Av. Caps" shows the available capacity on all flights. The number of possible delay lengths, which is the number of scenarios, is given in the fifth row, and in the sixth row is the possible delay lengths in these different scenarios.

Table 4.1: Data for Instances 1 to 4

Data Info	Ins 1	Ins 2	Ins 3	Ins 4
Num Pass	3	50	100	100
Num Fts	8	1144	1144	1144
Num Final Dests	3	15	15	30
Av. Caps	1	5	5	4
Num Scens	2	4	4	4
Delays	90, 300	30, 90, 210, 300	30, 90, 210, 300	30, 90, 210, 300

Table 4.2: Data for Instances 5 to 7

Data Info	Ins 5	Ins 6	Ins 7
Num Pass	100	200	200
Num Fts	1144	1144	1144
Num Final Dests	15	15	15
Av. Caps	8	10	12
Num Scens	16	16	40
Delays	30, 60,...,480	30, 60,...,480	10, 20,..., 400

For all of our instances except the first, we use the large data set described in section 4.2.1. The delayed flight is Flight 33, leaving at 8:20 am and traveling from JFK to ATL, on which we vary the number of passengers. Having 100 or 200 passengers is representative of some typical aircraft sizes. Also, we are interested in how the available capacities may affect the run time and the number of iterations in Benders decomposition, so we vary the available capacities from 4 to 12, except for instance 1. In practice, the number of scenarios may be quite large, especially when we later have more than one delayed flight as discussed in chapter V, so instances 5, 6, and

7 are particularly useful, especially in evaluating the run time. Note that in the first instance, we use a small data set with only 8 flights, also described in section 4.2.1.

Next, for each instance, we run several "simulations." In all simulations other than those for the first instance, we change two pieces of information. The first is the final destinations of the passengers on the delayed flight. There are 85 different locations, and we randomly select 15 or 30 of them to be the final destinations. This list must include the destination of the delayed flight, since we assume $\frac{1}{3}$ of the passengers do not have connecting flights. The rest of the passengers are evenly distributed over the other final destinations. For example, suppose we have 100 passengers and the final destinations are $2, 24, 66, 74, 36, 63, 51, 60, 73, 25, 30, 17, 41, 35$, and 84. Then 33 passengers are headed to location 2, and either 4 or 5 passengers are going to each of the other final destinations. The set of destinations changes the problem quite a bit, as different flights are used, the effects of available capacities change, and a different number of itineraries are available for each passenger.

The second piece of information that we change in each simulation is the order that the passengers are reaccommodated when using the current method. In each simulation, the order is randomly assigned for all the passengers that miss connections with their first-stage itineraries. In practice, the order passengers are assigned by airlines may often depend on if they are frequent flyers, are in business class, or if they are connecting to international flights. Randomly generating the order helps us understand situations that can occur, since we do not have this type of information on passengers. Thus, each instance is as described in the table above, and within each instance we run several simulations where we change only the final destinations of the passengers and the order that they are reaccommodated when using the current method.

Before we present the tables showing the results from running the instances, we describe the information given in the tables. In the "New AC" column, we have the average costs experienced per passenger using our method. Letting $\mathbf{x}^\star$ and $\mathbf{y}^{\omega\star}$ for all $\omega \in \Omega$ be the optimal solutions found, the entry for each scenario $\omega \in \Omega$ is the average realized cost in that scenario, which is

$$\frac{1}{|P|}\left(\sum_{p\in P}\sum_{i\in I_p}(1-\epsilon_i^\omega)c_{pi}x_{pi}^\star + \sum_{a\in A_e} c_a y_a^{\omega\star}\right) \tag{4.13}$$

Recall that we count c_{p1}, the cost of the itinerary involving only the delayed flight, as the delay d^ω in that scenario.

The first entry in the "New AC" column is the expected value for the average cost over all scenarios, which is the same as the objective value for our model divided by the number of passengers,

$$\frac{1}{|P|}\left(\sum_{p\in P}\sum_{i\in I_p}\left(1-\sum_{\omega\in\Omega}\epsilon_i^\omega\rho^\omega\right)c_{pi}x_{pi}^\star + \sum_{\omega\in\Omega}\rho^\omega\sum_{a\in A_e} c_a y_a^{\omega\star}\right) \tag{4.14}$$

Note that this is the weighted sum of the other entries in the column based on the probabilities for each scenario. To see this, consider the weighted average of the entries for each scenario.

$$\sum_{\omega\in\Omega}\rho^\omega\frac{1}{|P|}\left(\sum_{p\in P}\sum_{i\in I_p}\left(1-\epsilon_i^\omega\right)c_{pi}x_{pi}^\star + \sum_{a\in A_e} c_a y_a^{\omega\star}\right) \tag{4.15}$$

This can be written as

$$\frac{1}{|P|}\left(\sum_{p\in P}\sum_{i\in I_p}\sum_{\omega\in\Omega}\left(1-\epsilon_i^\omega\right)\rho^\omega c_{pi}x_{pi}^\star + \sum_{\omega\in\Omega}\rho^\omega\sum_{a\in A_e} c_a y_a^{\omega\star}\right) \tag{4.16}$$

Using that

$$\begin{aligned}\sum_{\omega\in\Omega}\Big(1-\epsilon_i^\omega\Big)\rho^\omega &= \sum_{\omega\in\Omega}\Big(\rho^\omega-\epsilon_i^\omega\rho^\omega\Big)\\ &= \sum_{\omega\in\Omega}\rho^\omega-\sum_{\omega\in\Omega}\epsilon_i^\omega\rho^\omega\\ &= 1-\sum_{\omega\in\Omega}\epsilon_i^\omega\rho^\omega\end{aligned}$$

the weighted average becomes the expression for the first entry in the column.

Next, in the "Cur. AC" column, we have the average cost experienced per passenger in the current method, where the delay experienced by any passengers not reaccommodated is not included. The first entry in the column is the average expected cost over all the scenarios, so it is the weighted sum of the rest of the entries in the column based on the probabilities for each scenario. We introduce some new notation to represent the entries more clearly.

- Let P_m be the set of passengers that miss a connection after taking the delayed flight and are then reaccommodated.
- Let I_p^ω be the set of possible itineraries for passenger $p\in P_m$ after missing a connection in outcome $\omega\in\Omega$.
- Let z_{pi}^ω be a binary variable that is 1 if passenger $p\in P_m$ is given itinerary $i\in I_p^\omega$ in outcome $\omega\in\Omega$.

Letting $\mathbf{z}^{\omega\star}$ be the optimal solution found in scenario $\omega\in\Omega$, the entry for each scenario $\omega\in\Omega$ in the column "Cur. AC" is

$$\frac{1}{|P|}\Big(|P_1|d^\omega+\sum_{p\in P_m}\sum_{i\in I_p^\omega}c_{pi}z_{pi}^{\star\omega}\Big) \tag{4.17}$$

Thus the first entry in the column is

$$\frac{1}{|P|}\sum_{\omega\in\Omega}\rho^{\omega}\Big(|P_1|d^{\omega}+\sum_{p\in P_m}\sum_{i\in I_p^{\omega}}c_{pi}z_{pi}^{\omega\star}\Big) \tag{4.18}$$

As mentioned, in some scenarios in many instances, some passengers cannot reaccommodated using the current method, and the average costs given in column "Cur. AC" do not represent the cost to these passengers. The number of passengers not reaccommodated is given in the column "Num NR." The first entry is the expected number not reaccommodated, and the other entries in the column are the numbers not reaccommodated in each scenario.

This situation creates an inaccurate comparison between the current method and the new method, since the cost for our method represents more passengers than for the current method. In the current method, the cost to passengers that do not reach their destinations is not counted. To provide a better comparison, we include a column "New AC r," in the charts, which represents "New average cost revised." Suppose that in some scenario $\omega \in \Omega$, n passengers are not reaccommodated in the current method. We subtract the n highest costs to different passengers using our method from the total cost in that scenario, then divide the result by $|P| - n$.

For example, suppose that in a certain scenario, our average cost is 186.81 and the current average cost is 175.12, where 23 passengers are not reaccommodated with the current method. The total cost using our method is $186.81 \cdot 100 = 18681$ if there are 100 passengers. We subtract the 23 highest passenger costs, which adds up to 6600, giving us $18681 - 6600 = 12081$. Thus the average cost, not counting the 23 highest costs, is $\frac{12081}{77} = 156.90$. This number is less than the average cost from the current method. Using the "New AC r" column, we thus can compare the cost from the two methods over the same number of passengers. Note that the "Cur. AC" data still has an advantage, since if n passengers are not reaccommodated at all with the current method, normally much fewer than n passengers are not reaccommodated with the new method.

Table 4.3 shows data collected from just one simulation of instance 3 to illustrate the information.

Table 4.3: Data from One Simulation of Instance 3

Data	New AC	New AC r	Cur. AC	Num NR
Exp.	96.44	83.60	127.07	8.5
ω=0	9.9	9.9	9.9	0
ω=1	36.99	36.99	48.01	0
ω=2	136.19	114.20	195.25	13
ω=3	202.69	173.33	255.11	21

Next, we have two tables for each instance. We ran 15 simulations of each instance, where the changes in the data are due to random generation of the final destinations of the passengers and the order that they are reaccommodated in for the current method. For each simulation, we have two rows, comparing the results for our method and the current method. Also, the first two rows give the average values over all the simulations. Here we report only one simulation and the averages for each instance, but the data from all simulations are given in appendix B. We next describe the data in each column in tables 4.4 through 4.9.

- The first data column, "EAC," is the expected average cost over all passengers and over all scenarios.
- The second data column, "EAC r," is the revised average expected cost for our method, as described earlier in this section. Note that for the rows for the current method, this column is not applicable.
- The column "END" represents the expected number of passengers delayed considering all scenarios.
- The column "END30" shows the expected number of passengers delayed more than 30 minutes.

- The fifth data column shows the percentage that the expected average cost revised for the new method is of the expected average cost for the current method.

- The sixth data column shows the expected number of passengers that do not reach their destinations by the end time.

- The column "EHC" shows the expected value of the highest passenger cost over all scenarios. This number shows a different perspective than the expected values do. Note that we have assigned a cost of 4320 minutes for not being able to be reaccommodated, which represents three days.

- Last, sometimes we have a column "ELC," which is the expected lowest cost experienced by any one passenger over all scenarios. If this column does not exist, then the expected lowest cost is 0.

Table 4.4: Instance 2

Sim	Method	EAC	EAC r.	END	END30	Cost%	ENNR	EHC
Ave	New	85.7	74.7	25.4	21.4	65%	0	255.2
Ave	Cur.	116.4		29.3	25.3		2.7	2231.2
1	New	92.8	80.2	27.5	23.5	62%	0	232.3
1	Cur.	128.4		31.5	27.5		4.25	2201

Table 4.5: Instance 3

Sim	Method	EAC	EAC r.	END	END30	Cost%	ENNR	EHC
Ave	New	96.5	72.1	51.8	43.6	63.4%	0.05	513.6
Ave	Cur.	114.0		57.3	49.1		10.6	2706.7
1	New	105.9	82.2	55.0	46.8	63%	0	412.5
1	Cur.	131.4		61.5	53.3		10.5	3247.5

Table 4.6: Instance 4

Sim	Method	EAC	EAC r.	END	END30	Cost%	ENNR	EHC
Ave	New	92.7	80.3	53.6	45.3	68%	0	371.0
Ave	Cur.	119.1		58.4	50.1		5.5	2683.3
1	New	104.2	93.6	60	51.8	68%	0	350.5
1	Cur.	138.2		64.3	56		5.8	2326.5

Table 4.7: Instance 5

Sim	Method	EAC	EAC r.	END	END30	Cost%	ENNR	EHC	ELC
Ave	New	159.4	110.2	63.8	61.8	54%	0	390.1	0.0
Ave	Cur.	204.0		72.0	69.8		18.7	3681.7	21.9
1	New	166.7	114.3	64.1	62.1	51%	0	445	0
1	Cur.	223.4		75.4	73.3		18.8	4052	0

Table 4.8: Instance 6

Sim	Method	EAC	EAC r.	END	END30	Cost%	ENNR	EHC
Ave	New	157.9	89.9	126.0	121.2	50%	0.1	645.1
Ave	Cur.	178.3		136.3	132.1		49.4	3739.5
1	New	201.1	94.2	123.6	119.4	55%	1.9	3817
1	Cur.	171.4		133.8	129.7		49.4	4052

Table 4.9: Instance 7

Sim	Method	EAC	EAC r.	END	END30	Cost%	ENNR	EHC
Ave	New	123.9	78.6	116.2	111.3	52%	0	330.6
Ave	Cur.	152.2		134.0	128.9		42.3	3758.9
1	New	144.2	85.6	127.2	122.3	47%	0	491
1	Cur.	182.2		147.9	142.9		49.8	4105

In tables 4.10 through 4.15, we present the same information, but specifically for the scenario with the highest delay. Note that we do not have a column for the number of passengers delayed more than 30 minutes, because the values were the same as the number delayed. Also, if the value reported in the column “HC” for the highest passenger cost is “No Arr,” the worst cost comes from not arriving by the end time.

Table 4.10: Instance 2, $\omega = 4$

Sim	Method	AC	AC r.	ND	ND30	Cost%	ENNR	HC
Ave	New	165.9	137.2	31.2	31.2	57%	0	385.5
Ave	Cur.	244.1		41.8	41.6		7.3	No Arr.
1	New	188.1	155.2	34	34	59%	0	355
1	Cur.	264.6		44	44		10	No Arr.

Table 4.11: Instance 3, $\omega = 4$

Sim	Method	AC	AC r.	ND	Cost%	ENNR	HC
Ave	New	181.3	130.6	63.5	58.7%	0.07	444.6
Ave	Cur.	221.6		77.5		23.5	No Arr.
1	New	194.9	147.9	70	60%	0	540
1	Cur.	246.9		85		23.0	No Arr.

Table 4.12: Instance 4, $\omega = 4$

Sim	Method	AC	AC r.	ND	ND30	Cost%	ENNR	HC
Ave	New	179.8	150.3	67.2	66.9	63%	0	475.6
Ave	Cur.	239.7		81.0	80.6		14.3	No Arr.
1	New	200.4	177.9	77	77	66%	0	436
1	Cur.	270.2		88	88		13	No Arr.

Table 4.13: Instance 5, $\omega = 16$

Sim	Method	AC	AC r.	ND	Cost%	ENNR	HC	LC
Ave	New	273.8	122.4	68.0	31%	0	521.6	0.0
Ave	Cur.	398.7		91.4		42.4	No Arr.	102.9
1	New	278.3	102.5	66	24%	543	0	0
1	Cur.	421.5		94		45	No Arr.	0

Table 4.14: Instance 6, $\omega = 16$

Sim	Method	AC	AC r.	ND	ND30	Cost%	ENNR	HC
Ave	New	273.3	93.8	136.5	135.9	25%	0.1	780.3
Ave	Cur.	367.3		176.4	176.4		94.3	No Arr.
1	New	301.1	91.6	128	128	27%	2	No Arr.
1	Cur.	335.2		167	167		86	No Arr.

Table 4.15: Instance 7, $\omega = 40$

Sim	Method	AC	AC r.	ND	Cost%	ENNR	HC
Ave	New	241.1	117.2	134.5	38%	0	478.7
Ave	Cur.	312.0		174.5		83.9	No Arr.
1	New	249.2	108.8	136	30%	0	515
1	Cur.	360.6		186		93	No Arr.

In table 4.16, we give the information for instance 1. There was only one simulation run, because there is no randomness in the data set. We do not have a column for our average expected cost revised, because the data is the same as the column for our average expected cost. In addition, the number delayed more than 30 minutes is the same as the number delayed. Last, in table 4.17, we present data on the number of passengers given itineraries in the first-stage problem that do not include the delayed flight using the new method. In the first instance, one passenger was given one of these itineraries. Data from all simulations are given in appendix B.

Table 4.16: Instance 1

Data	Method	EAC	END	Cost%	ENNR	EHC
All ω	New	65	1	65.0%	0	195
All ω	Cur.	100	1.5		0	195
$\omega = 1$	New	100	1	58.8%	0	300
$\omega = 1$	Cur.	170	2		0	300

Table 4.17: Passengers moved off delayed flight in first stage

	Ins 2	Ins 3	Ins 4	Ins 5	Ins 6	Ins 7
Ave	12.7	17.1	17.4	37.1	69.9	52.1
Sim 1	12	15	15	43	62	66

4.2.4 Evaluation and Comparison of Methods

Having simulated the two different approaches to reaccommodating airline passengers and demonstrating through the computational results in section 4.2.3 that our proposed method yields better passenger outcomes in our instances, we now analyze the differences in delays and the quality of the solutions.

There is a significant range in the average delays experienced, which is not surprising. Some destinations are much harder to get to than others, based on which markets have many flights going to and from them. As expected, our revised average expected cost is significantly less than the current average expected cost and the current average cost in the worst scenario in all simulations.

We have done some statistical analysis on the data for all the instances. First is a hypothesis test comparing our revised average expected cost and the current average expected cost. Letting μ_c and μ_n be the average expected costs for the current method and new method, respectively, our null hypothesis is $H_0 : \mu_c - \mu_n = 0$. Thus our null hypothesis says that there is no difference between μ_c and μ_n. The alternative hypothesis is $H_1 : \mu_c - \mu_n > 0$. Note that this is a one-sided alternative hypothesis. Since our method has the added benefits of allowing passengers to get off the delayed flight and considering all passengers together, the cost from our method will never be more than the cost from the current method. Any solution that can be achieved using the current method can also be achieved using our method.

The t-statistic is $t = \frac{\bar{y}}{s/\sqrt{n}}$, where $\bar{y}$ is the average value of $\mu_c - \mu_n$ over the simulations, s is the sample standard deviation of the value of $\mu_c - \mu_n$ in the simulations, and n is the number of simulations. The t-statistic values are given in table 4.18. Using these values, for all instances, the null hypothesis can be rejected even at the 99.95% confidence level, since the critical value is then 4.1405.

Table 4.18: t values

Ins 2	Ins 3	Ins 4	Ins 5	Ins 6	Ins 7
11.71	15.04	27.78	17.96	16.81	14.03

In addition, we have created a 95% confidence interval for the cost percentage in each scenario, $[\bar{y} - 2.145\frac{s}{\sqrt{n}}, \bar{y} + 2.145\frac{s}{\sqrt{n}}]$. The intervals are given in table 4.19.

Table 4.19: 95% confidence intervals for the cost percentage

	Ins 2	Ins 3	Ins 4	Ins 5	Ins 6	Ins 7
Lower bound	61.0%	59.3%	65.7%	50.0%	45.8%	46.5%
Upper bound	68.5%	67.4%	69.3%	58.0%	55.4%	57.7%

It is important that the highest delay experienced for any one passenger decreases significantly in most simulations in all instances, both in expectation and in the scenario with the highest delay. The only exception is the eleventh simulation for instance 3, where the highest expected cost for the new and current methods are 3276.5 and 3247.5, a difference of 29 minutes. In this simulation, all of the other measures of the solution are better for the new method.

The number of passengers delayed and the number delayed over 30 minutes decreases as well while using the new method. It is notable that in every simulation, all of the values both in expectation and in the worst scenario are better using the new method. The only exception is a couple of simulations, where the expected number of people delayed is slightly higher or the same using the new method, since the average delay length is minimized instead, but the other values are significantly better with the new method.

Note that if the capacity is big enough, such as the same as the number of passengers under consideration, then the only advantage our model has over the current method is that we can preemptively move passengers off the delayed flight so that they do not miss connections. For any passengers that do have to be given new itineraries after the first stage, considering everyone together versus one-by-one will give the same quality of solution, since passengers are not competing for seats at all.

We have presented results from implementing our model and analyzed the data showing the effectiveness of our method. We conclude that considering this method would be beneficial for an airline. First, it may help the airline to lower costs, since the available capacity on flights can be best utilized. Also, more employees are needed if passengers are reaccommodated one-by-one after missing connections, but not as many may be necessary with an automated decision-making tool for reaccommodation. In addition, airlines may be fined in certain situations involving reaccommodation, and avoiding these fines is a priority. Last, better handling the reaccommodation of passengers will produce happier customers that want to fly with the airline again in the future, which can improve profits. Implementing our model may require more technical development and devices, but the initial cost should be minimal compared to the benefits.

4.2.5 Quality of Solutions and Cost of Delay

We have implemented the model from section 3.2.6 and evaluated the computational results in several instances, finding that our method significantly reduces passengers' delays and would be beneficial to airlines. There are several considerations about the quality of our solutions and how much they improve passengers' experiences. We have assumed a linear relationship between the delay a passenger experiences and the cost that should be assigned for that delay. For example, we assume that a four hour delay is twice as unpleasant as a two hour delay. In reality, a longer delay may allow a passenger to go back home and come back later, rather than waiting in the airport for a few hours. A passenger may miss a meeting with a two hour delay but not with a one hour delay, so the two hour delay is much more than twice as frustrating. Also, it may be more accurate to assign a higher cost to one passenger being delayed two hours than two passengers waiting one hour each.

Individual passengers have different preferences, since the level of frustration depends on the cause of the travel, the location of home, and other factors. Unfortunately, we cannot know people's preferences. Some may prefer a 24 hour delay to a 3 hour delay, while others would not. How accurate are our solutions for producing the least inconvenience? We argue that our objective function is a good substitute, especially since it is not possible to poll the passengers. Minimizing total delay is certainly a goal passengers would support and definitely produces valuable results. Most passengers are probably happy with simply arriving as soon as possible.

Note that when there is stochasticity in the delay length, it is possible to give a passenger a new itinerary that causes a higher delay than if he or she had kept the original itinerary. For example, suppose that a passenger has a two-flight itinerary consisting of a flight from JFK to ATL leaving at 8:20 am and arriving at 11:08 am and then a flight to LAS leaving at 2:30 pm and arriving at 7:30 pm. Depending on the chance of the passenger missing her connection, our solution may say to move the person to a two-flight itinerary arriving at LAS at 10:00 pm. This solution is best in expectation but it could turn out that the delay is short enough that the passenger could have made her connection.

How can we handle this possibility? One idea is to assign a higher weight to alternative itineraries that arrive later than the passenger's original one does in the best-case scenario. Second, we can make a rule that no first-stage itinerary can be assigned that arrives a certain amount of time later than the original one does in the best case. Third, we can give passengers the option of refusing a change to a different flight in the first-stage solution. After passengers take their first-stage itineraries, the second-stage problem is then re-solved over all passengers that were assigned the delayed flight and miss a connection plus all passengers that were not assigned the delayed flight but decided to keep it anyways and then miss a connection.

4.3 Run time

In the previous section, we discussed the efficacy of our approach and showed how using the model from section 3.2.6 can significantly reduce the length of delays passengers experience. Now in this section, we evaluate the tractability of our approach.

Specifically, how fast can the model be solved in different situations? Is it fast enough that it can be useful for an airline when facing a delay?

We discuss solving our model in section 3.2.6 in two different ways. The first way is to solve the complete mixed-integer model all at once, and the second is to use Benders decomposition to break down the model as explained in section 4.1. We present data from different instances and analyze the run time, comparing the two methods of solving the model and different factors that influence the run time.

Note that, while doing one iteration of Benders decomposition, we have to solve $|\Omega|$ subproblems, one after the other. In actuality, the subproblems' solutions are not related to each other, so they could be solved all at the same time on different machines. In that case, the time to solve all of them is the highest time to solve any one of them. We compare the run times using this perspective of parallelization of the algorithm as well.

We also discuss the number of iterations completed while using Benders decomposition in different instances and simulations. We look at what affects that number, including properties of the data set of flights, delay scenarios, and the available capacities of the non-delayed flights.

4.3.1 Results

We present results on the run time from different instances. The data about these instances are described in the table in section 4.2.3. For each simulation of each instance, we change only the passengers' final destinations, as described more fully in section 4.2.2. We present the average over 15 simulations for each instance and show data from one simulation. The data from all simulations is given in appendix C. We now describe the data given in tables 4.20 through 4.26, where all times are in seconds.

- The first data column gives the number of flights that are part of a possible itinerary for at least one passenger. This shows the size of the data set that is actually part of the model.
- The second column shows the total time it take to create and solve the model as a mixed-integer program all at once.
- The third column shows the total time to create and solve the problem using Benders decomposition.
- The column "Bend, p," representing "Benders decomposition time, parallelization," gives the total time to create and solve the model using Benders decomposition if we parallelize the algorithm.
- The column "B/MIP" gives the percentage that the time for Benders decomposition is of the time to solve the model as a mixed-integer program.
- The second to last column gives the percentage that the time for Benders decomposition is of the time to solve the problem as a mixed-integer problem if we parallelize the algorithm.
- The last column "# Its" shows the number of iterations completed while using Benders decomposition.

In instance 1, only one simulation is recorded, since there is no randomness in the data set. For instances 6 and 7, it is not possible to solve the model without Benders decomposition, because the memory is exceeded, so no data can be reported on this topic. Recall that all run times are reported in seconds.

Table 4.20: Instance 1 run time

# Fts	MIP	Bend	Bend, p	B/MIP	B,p/MIP	# Its
6	0.06	0.44	0.34	703%	553%	2

Table 4.21: Instance 2 run time

Sim	# Fts	MIP	Bend	Bend, p	B/MIP	B,p/MIP	# Its
Ave	96.3	7.6	10.6	6.3	141%	83%	2.0
1	103	7.2	10.3	6.2	143%	86%	2

Table 4.22: Instance 3 run time

Sim	# Fts	MIP	Bend	Bend, p	B/MIP	B,p/MIP	# Its
Ave	100.5	7.0	19.6	10.3	282%	147%	4.9
1	68	6.7	28.2	14.2	423%	214%	7

Table 4.23: Instance 4 run time

Sim	# Fts	MIP	Bend	Bend, p	B/MIP	B,p/MIP	# Its
Ave	155.2	7.9	11.9	7.7	150%	97%	2.1
1	164	8.1	16.0	12.2	198%	151%	2

Table 4.24: Instance 5 run time

Sim	# Fts	MIP	Bend	Bend, p	B/MIP	B,p/MIP	# Its
Ave	94.9	28.0	36.6	14.4	131%	52%	2.1
1	94	27.0	36.9	15.7	137%	58%	2

Table 4.25: Instance 6 run time

Sim	# Fts	MIP	Bend	Bend, p	# Its
Ave	92.2	n/a	138.3	62.7	6.6
1	75	n/a	124.0	55.1	6

Table 4.26: Instance 7 run time

Sim	# Fts	MIP	Bend	Bend, p	# Its
Ave	94.2	n/a	186.7	87.1	3.2
1	92	n/a	483.7	256.0	7

4.3.2 Benders Decomposition

In order for an airline to implement the new method, the models need to be quick enough to solve so that the solution can be found in a reasonable amount of time. The tables in section 4.3.1 show the differences in the run time while using Benders decomposition versus solving the same problem explicitly as a mixed integer program, and we are interested in how using Benders decomposition affects the solve times.

We see that the effect on the run time depends on the instance. For smaller models, such as instance 1 through 4, using Benders decomposition does not improve the run time as much as it does for larger models. For instance 5, we see that the average value in the column "B,p/MIP" is 52%, which is a significant improvement. In instances 6 and 7, the models cannot even be solved without using Benders decomposition with the machine we used. Table 4.27 illustrates the differences in the size of the model in instances 5, 6, and 7 averaged over four simulations.

Table 4.27: Size of Model

Instance	# Variables	# Constraints
4	1,555,735	6,518
5	1,783,178	7,949
6	4,449,606	19,160

It is interesting that in instance 3, using Benders decomposition actually increases the run time by 57% on average, even while solving the subproblems on different machines. Assuming a model can be solved without Benders decomposition, the data show that if more iterations must be completed while doing Benders decomposition, then it does not help the run time much, if at all. Note that for instance 3, the average number of iterations completed is 4.1, while for instances 2, 4, and 5, the average is about 2.

Parallelizing the algorithm has a profound effect on whether or not using Benders decomposition is beneficial. For example, in instance 2, the average for the column "B/MIP" is 141%, so using Benders decomposition increases the run time by 41%. On the other hand, the average of the column "Bend,p/MIP" is 83%, so using Benders decomposition is helpful.

4.3.3 Factors and Conclusions

We have presented results showing the run time and have seen how using Benders decomposition affects the solve time in various situations. There are several differences in the instances shown in the tables in section 4.2.3. We now discuss the run times and different factors affecting them while using Benders decomposition or solving the whole mixed-integer problem all at once. We also look at how the factors affect the number of iterations completed while using Benders decomposition.

First, we consider the effect of the number of passengers from the delayed flight on the run time and number of iterations. In instance 2 there are 50 passengers, and in instance 3 there are 100 passengers, and this is the only differences between these instances. The average time to solve the MIP model is close in the two instances. On the other hand, using Benders decomposition, the solve time and the number of iterations approximately double as the number of passengers doubles. It is difficult to determine exactly how the run time and number of iterations of Benders decomposition behave in general depending on the number of passengers, since other factors, such as infeasibility, may come into play. We suggest this as a topic of future research.

As the available capacities on the flights increase, the run time and number of Benders decomposition iterations decrease. This makes sense, since as the capacities increase, even approaching the number of passengers, more and more capacity constraints can be ignored since there are not enough passengers with those flights in their list of possibilities. In our instances, increasing the number of final destinations that the passengers can have is similar to increasing the available capacities, since the passengers are distributed over the final destination options. Thus, the run time also decreases as the number of final destinations increases.

Another factor clearly contributing to the run time is the number of scenarios for the length of the delay, particularly when we do not parallelize the algorithm. As the number of scenarios increases, so does the run time. In instance 7, which has 40 scenarios, the run time jumps dramatically from the instances with only 16, 4, or 2 scenarios when the algorithm is not parallelized.

Note that parallelizing the algorithm has a profound effect on the run times while using Benders decomposition. On average, the percentage of time it takes to solve the problem if we parallelize the algorithm compared to the time if not solving it this way ranges from 39% to 65%, not including the first instance. This is definitely a method worth using in the future.

The number of flights in the data set also has an impact on the run time and number of iterations. A bigger data set corresponds to a higher run time, holding all other factors constant, since the number of constraints and variables is higher. If we double the number of flights, then the number of itineraries in the first-stage problem and the number of arcs in the second-stage problem increase by more than one hundred percent. Another important factor is not just the number of flights in the data set, but the number of flights that are actually part of a possible itinerary for at least one passenger. This number is often significantly less than the number of flights, as shown in tables 4.20 through 4.26, since passengers are going to only so many different destinations.

It is significant that the run times are relatively quick, even for instance 7 with 200 passengers and 40 scenarios. The average time to solve it using Benders decomposition while parallelizing the algorithm is less than 1.5 minutes. The simulation that took the longest ran for 4 minutes and 16 seconds. Even this is quick enough to be used by an airline in practice. For the other instances, the run times were often several seconds.

A factor contributing to the relatively short run times is that the number of itineraries in both stages is often small. In section 3.1.4, we showed that the number of itineraries in the first-stage is $O(F^3P)$, but that the number was often small. For example, in the second table in section 3.1.4, when there is 100 passengers, 91 possible flights, and 15 different final destinations, the number of variables in the one-stage itinerary-based model is 2237. This is $2237/100 = 22.37$ on average over all the passengers. Since an itinerary can have only up to three flights that must arrive before the end of the day, the number is limited.

Recall that the MIP model has integral first-stage variables and the restricted master problem has all integral variables. Fortunately, while solving these models, Cplex often does not have to branch and is able to find the solution at the root node by using standard pre-processing, making the programs solve much faster. The pre-processing is able to eliminate many variables and constraints, and the model has a good structure that tends towards integral solutions. Occasionally, while solving the restricted master problem in some simulations, Cplex has to spend a couple of minutes branching to find an optimal integral solution. Sometimes it reaches the 1000'th node, but this is not common. It is significant that we do not need to branch to all possible nodes or consider all possible solutions while branching. The solution of the linear relaxation brings us to the general region where the solution is in the branch and bound tree, and branching allows us to choose an integer solution in that region.

CHAPTER V

Multiple Flights Delayed

In chapters III and IV, we considered the case where only one flight is delayed. In reality, flights are often delayed concurrently. For example, if there is a snowstorm or other bad weather, many flights may be affected. In addition, a delay in one flight, even as a result of a mechanical problem, could cause delays in other flights. Two aircraft may be supposed to occupy the same gate, but one of them cannot because the other aircraft is still there because of a delay. In addition, a flight may intentionally be delayed to await incoming connecting passengers from another delayed flight. If there is a security problem at a specific airport, many flights departing from that airport may be delayed. Thus concurrent delays are possible, both in the case of flights taking off at times close to each other and farther apart, and between flights at the same airport or different locations.

The goal of reaccommodating passengers becomes much more complicated when multiple flights are delayed. Passengers from one flight may be moved to other flights which are then themselves delayed, so the possibilities for the delay of each flight affect the decisions for all passengers, and there are many more scenarios to consider. To model the problem with more than one delayed flight, we use a different type of formulation, employing the idea of a "portfolio" of flights for each passengers. We first describe the notion of a portfolio, then present the portfolio model, and discuss methods for solving it. We give some examples along the way, and last discuss having more than two delayed flights.

5.1 The Portfolio Model for Two Delayed Flights

5.1.1 Definition of a Portfolio

We use a different type of formulation to model this situation with more than one flight delayed, where we assign a "portfolio" of flights to each passenger. This portfolio determines which flights the passenger should take from the current location. It also determines which flights the passenger should take in each possible scenario in which she would be disrupted if it occurs. Thus, we decide all at once the decisions for whatever ends up happening in the system. Note that the second-stage part of some of the portfolios depends on the outcome for both delayed flights, so some passengers may have many different second-stage itineraries in their assigned portfolios.

Before solving the model, we create a list of all the possible portfolios for each passenger $p \in P$, so we know what flights are in each portfolio and the cost of each portfolio. We assume that we can find the set of possible first-stage itineraries I_p for passenger p, which is made up of all itineraries that arrive at the desired destination in at least one scenario by the end time. For each $i \in I_p$ and each scenario $\omega \in \Omega$ where the passenger misses a connection with itinerary i, there is a list of possible second-stage itineraries I_p^ω based on the time of day and location in the scenario after the delayed flight has been taken and the connection missed. For these scenarios where misconnections occur, we create all combinations for possible second-stage itineraries.

For example, suppose there are three scenarios. In the second and third, the passenger p will miss a connection with a certain first-stage itinerary $i_1 \in I_p$. In scenario $\omega = 2$, there are three second-stage itineraries as options, $a, b, c \in I_p^2$, and in scenario $\omega = 3$, there are only two of these options $b, c \in I_p^3$, since it is later in the day. Then the possible portfolios with first-stage itinerary i_1 are shown in table 5.1.

Table 5.1: Portfolios with First-stage Itinerary i_1

	Itinerary 1			Itinerary 2			Itinerary 3		
1st stage	i_1			i_1			i_1		
2nd stage	$\omega = 1$	$\omega = 2$	$\omega = 3$	$\omega = 1$	$\omega = 2$	$\omega = 3$	$\omega = 1$	$\omega = 2$	$\omega = 3$
	n/a	a	b	n/a	a	c	n/a	b	b

	Itinerary 4			Itinerary 5			Itinerary 6		
1st stage	i_1			i_1			i_1		
2nd stage	$\omega = 1$	$\omega = 2$	$\omega = 3$	$\omega = 1$	$\omega = 2$	$\omega = 3$	$\omega = 1$	$\omega = 2$	$\omega = 3$
	n/a	b	c	n/a	c	b	n/a	c	c

The number of portfolios in different problems is explored in section 5.2.2. Before we show the formulation, we have a list of comments to clarify the possibilities for portfolio choices. Also, a helpful example of a passenger's set of possible portfolios is given in section 5.1.4.

- It is permissible to give a passenger a delayed flight that he or she was not originally taking in the first-stage problem.
- A passenger can be assigned more than one delayed flight in the first stage if the flights make up a viable itinerary in at least one scenario.
- A passenger can be assigned a delayed flight in the second-stage problem only if there is no chance of missing a connection, since we do not have a third-stage problem. This is possible only in some situations. Note that we did not have a rule like this one in the case with only one delayed flight. Since all passengers were at the origin of the delayed flight, no passengers could be assigned the delayed flight in the second stage anyway.

5.1.2 Discussion of Stages

In a portfolio, a passenger's first-stage flights are the ones given when the delay of the passenger's flight is discovered. Any second-stage flights are those taken after a disruption with the first-stage itinerary. We now make an assumption that there are two delayed flights, and an idea needs clarification. Since the flights' timing may differ, we need to know what second-stage solution to use when one of the flights lands. The arguments we present can be expanded for when more than two flights are delayed.

When we reaccommodate passengers from two delayed flights using just one model, the outcome that occurs and the second-stage solution depend on both delayed flights. Suppose that one delayed flight f_a lands before the length of delay of the other flight f_b is known. Decisions for disrupted passengers from f_a must be made right away, so they do not miss flight options. Unfortunately, we may not know what scenario is occurring, since it depends on f_b, so we do not know what solution to implement.

The best answer to this problem is to add the rule before solving the model that the second-stage solution for the passengers on flight f_a must be the same no matter what the delay is for f_b in this type of situation. After flight f_a lands, we then know how to reaccommodate any disrupted passengers. When creating the lists of possible portfolios for the passengers, we do not include any portfolios that do not satisfy this rule. Even though we solve the model as a two-stage process, we may execute it in a multi-stage process as the different flights land and have disrupted passengers at different times of the day. Adding this rule may not necessarily give the optimal solution in some outcomes, but the solutions will be quite good, feasible, and executable.

For example, consider two flights 100 and 200 scheduled to depart at 12pm and 12:30pm, respectively. The chart 5.2 shows the possible delay lengths, departure times, and arrival times for the flights in different scenarios.

Table 5.2: Scenarios for Flights 100 and 200

Scenario	100 delay	100 dept	100 arr	200 delay	200 dept	200 arr
$\omega = 1$	20	12:20	1:20	20	12:50	2:20
$\omega = 2$	30	12:30	1:30	20	12:50	2:20
$\omega = 3$	20	12:20	1:20	40	1:10	2:40
$\omega = 4$	30	12:30	1:30	120	2:30	4:00
$\omega = 5$	**60**	1:00	2:00	**120**	2:30	4:00
$\omega = 6$	**60**	1:00	2:00	**180**	3:30	5:00
$\omega = 7$	90	1:30	2:30	90	2:00	3:30
$\omega = 8$	90	1:30	2:30	100	2:10	3:40
$\omega = 9$	90	1:30	2:30	150	3:00	4:30

We have three situations to consider, where we assume that the delay length of a flight becomes known at the same time that it departs.

- In scenarios $\omega = 1$ and $\omega = 2$, Flight 200 has a delay of 20 minutes, while Flight 100 has a delay of 20 or 30 minutes. In this case, we do not need to add any rules. To see this, in either scenario, the delay length for 100 becomes known before 200 departs, so the scenario that occurs is clear.

- In scenarios $\omega = 5$ and $\omega = 6$, Flight 100 has a delay of 60 minutes, and Flight 200 has a delay of either 120 or 180 minutes. In this situation, the delay length for Flight 200 becomes known at 2:30, and Flight 100 lands at 2:00. We add a rule that the second-stage itineraries for any disrupted passengers taking Flight 100 must be equivalent in scenarios $\omega = 5$ and $\omega = 6$.

- In scenarios $\omega = 7$, $\omega = 8$, and $\omega = 9$, the length of delay for 100 is 90 minutes, but for 200, it can be either 90, 100, or 150 minutes. Fortunately, when 100 arrives at 2:30, we know the length of delay for 200, possibly by the method of elimination. If the length is 90 or 100 minutes, then 200 takes off before 100 arrives, so we know what solution to implement. If 200 has not departed by the time that 100 lands, then we know scenario $\omega = 9$ occurs. Thus, no rules need to be added for these three scenarios.

5.1.3 Explanation of the Portfolio Model

Next, we introduce the Portfolio Model and some notation. Recall that we assume there are two delayed flights f_A and f_B. A list and description of all of the sets, parameters, and variables for the model are also given in appendix A. First, the sets:

- Let P be the set of passengers on either of the two delayed flights.
- Let Q_p be the set of possible portfolios for passenger $p \in P$.
- Let Ω be the set of possible pairs of delays of flights f_A and f_B.

Next we have the parameters for the model:

- Let c_{pq} be the expected cost (length of delay) of portfolio $q \in Q_p$ for passenger $p \in P$. This is calculated as $\sum_{\omega \in \Omega} \rho^{\omega} c_{pq}^{\omega}$, where c_{pq}^{ω} is the length of delay in outcome $\omega \in \Omega$.
- Let C_f be the available capacity on flight $f \in F$.

Last, we have a description of the variables for the model:

- Let ϵ_{qf}^{ω} be 1 if there is a misconnection in the first stage in portfolio $q \in Q_p$ causing flight $f \in F$, which is part of q, to be missed in outcome $\omega \in \Omega$.
- Let δ_{qf} be 1 if flight $f \in F$ is in the first-stage part of portfolio $q \in Q_p$, and let it be 0 otherwise.
- Let δ_{qf}^{ω} be 1 if flight $f \in F$ is in the second-stage part of portfolio $q \in Q_p$ in outcome $\omega \in \Omega$, and let it be 0 otherwise.
- Let z_{pq} be 1 if passenger $p \in P$ is given portfolio $q \in Q_p$ and 0 otherwise. This is the only type of decision variable in the model.

The portfolio model is given below.

$$
\begin{aligned}
&\text{minimize} \sum_{p\in P}\sum_{q\in Q_p} c_{pq} z_{pq} && && (5.1)\\
&\text{subject to} \sum_{q\in Q_p} z_{pq} = 1 && \forall p \in P && (5.2)\\
&\sum_{p\in P}\sum_{q\in Q_p} \delta_{qf} z_{pq} \leq C_f && \forall f \in F && (5.3)\\
&\sum_{p\in P}\sum_{q\in Q_p} \Big(\delta^{\omega}_{qf} + \delta_{qf}(1-\epsilon^{\omega}_{qf})\Big) z_{pq} \leq C_f && \forall f \in F,\, \omega \in \Omega && (5.4)\\
&z_{pq} \in \{0,1\} && \forall p \in P,\, q \in Q_p && (5.5)
\end{aligned}
$$

The first set of constraints says that every passenger must be assigned exactly one portfolio. The second set enforces that the capacities on the flights are not exceeded in the first-stage problem. The third set says that the sum of the passengers given a flight f in the second stage in a certain scenario ω plus the passengers assigned f in the first-stage solution who do not get disrupted cannot exceed the capacity of the flight. Last, the variables are binary.

For the objective function, we sum the cost of the chosen portfolios over all passengers. Note that for a portfolio $q \in Q_p$ that has a chance of disruption for passenger $p \in P$, the cost c_{pq} is not known for sure. We let c_{pq} be the expected cost, so $c_{pq} = \sum_{\omega\in\Omega} \rho^{\omega} c^{\omega}_{pq}$, where c^{ω}_{pq} is the cost of the portfolio in outcome ω and ρ^{ω} is the probability of outcome $\omega \in \Omega$ occurring.

5.1.4 Example of the Portfolio Model

We illustrate the idea of a portfolio and the portfolio model with an example that has two delayed flights and one passenger. Suppose the passenger p is at location PDX with an itinerary consisting of the following two flights.

- Flight 33 going to location LAX, scheduled to leave at 8:20 am and land at 11:08 am
- Flight 41 to location ATL scheduled to leave at 2:00 pm and arrive at 7:00 pm.

There are two delayed flights that are relevant to the passenger.

- Flight 33 has a delay of one hour or three hours, so the passenger may miss her connection if she keeps Flight 33.
- Flight 9 from location LAX to ATL, scheduled to leave at 1:00 pm and arrive at 6:00 pm, is delayed 30 minutes or 2.5 hours.

Note that $\Omega = \{(60, 30), (60, 150), (180, 30), (180, 150)\}$, where the first coordinate is the delay in Flight 33 and the second is the delay of Flight 9. We let the probability of each scenario be 0.25. All flights that are relevant to the passenger are depicted in table 5.3 and network 5.1. Note that the scheduled departure and arrival times are given in the table, even though Flights 33 and 9 are delayed. The connections shown in the network correspond to no delays.

Table 5.3: Relevant Flights

Flight Info	Ft 33	Ft 57	Ft 9	Ft 41	Ft 45	Ft 66
Origin City	PDX	PDX	LAX	LAX	LAX	PDX
Dest. City	LAX	LAX	ATL	ATL	ATL	ATL
Dept. Time	8:20 AM	9:55 AM	1:00 PM	2:00 PM	5:00 PM	12:00 PM
Arrival Time	11:08 AM	1:10 PM	6:00 PM	7:00 PM	10:00 PM	7:00 PM
Poss. Delays	60 180		30 150			

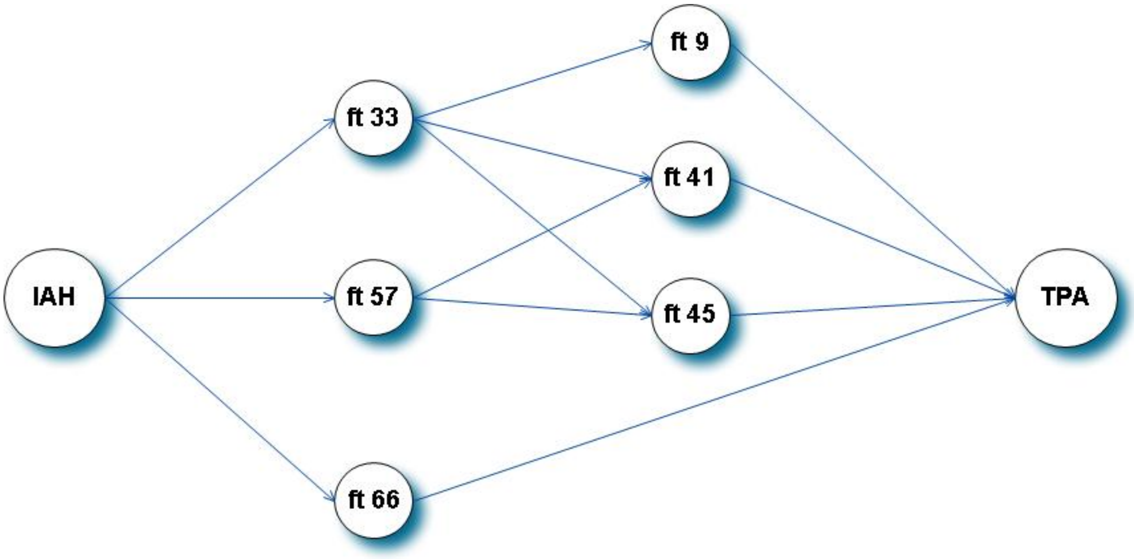

Figure 5.1: Flight Network

These flights make up the eleven different possible portfolios described in tables 5.4 and 5.5. The cost is the number of minutes later than 7:00 pm that the passenger will arrive.

Table 5.4: Some Possible Portfolios

Portfolio 1
Plan to take Flight 33 and Flight 9. Connection missed under scenario 3 with delays $(180, 30)$. In this case, take Flight 45 instead. The costs in the four scenarios are $0, 90, 180, 90$.
Portfolio 2
Plan to take Flight 33 and Flight 41. Connection missed under scenario 3 and 4 with a delay of 180 in Flight 33. In scenario 3, take Flight 45 instead. In scenario 4, take Flight 9 instead. The costs in the four scenarios are $0, 0, 180, 90$.
Portfolio 3
Plan to take Flight 33 and Flight 41. Connection missed under scenario 3 and 4 with a delay of 180 in Flight 33. In either case, take Flight 45 instead. The costs in the four scenarios are $0, 0, 180, 180$.

Table 5.5: More Possible Portfolios

Portfolio 4
Plan to take Flight 33 and Flight 45. The connection will not be missed. The cost is 180.
Portfolio 5
Plan to take Flight 57 and Flight 9. Connection missed under scenarios 1 and 3 with a delay of 30 in Flight 9. In either case, take Flight 41 instead. The costs in the four scenarios are $0, 90, 0, 90$.
Portfolio 6
Plan to take Flight 57 and Flight 9. Connection missed under scenarios 1 and 3 with a delay of 30 in Flight 9. In scenario 1, take Flight 41 instead. In scenario 3, take flight 45 instead. The costs in the four scenarios are $0, 90, 180, 90$.
Portfolio 7
Plan to take Flight 57 and Flight 9. Connection missed under scenarios 1 and 3 with a delay of 30 in Flight 9. In scenario 1, take Flight 45 instead. In scenario 3, take flight 41 instead. The costs in the four scenarios are $180, 90, 0, 90$.
Portfolio 8
Plan to take Flight 57 and Flight 9. Connection missed under scenarios 1 and 3 with a delay of 30 in Flight 9. In either case, take Flight 45 instead. The costs in the four scenarios are $180, 90, 180, 90$.
Portfolio 9
Plan to take Flight 57 and Flight 41. The connection will not be missed. The cost is 0.
Portfolio 10
Plan to take Flight 57 and Flight 45. The connection will not be missed. The cost is 180.
Portfolio 11
Plan to take Flight 66. The cost is 0.

In this example, Q_p consists of portfolios 1 through 11. To represent the passenger's portfolios, we use variables z_1 through z_{11}, instead of $z_{p,1}$ through $z_{p,11}$, since we are considering only one passenger. Since the passenger is not competing with other passengers for seats, and we assume that there is available capacity on each of the flights, we can assign whichever portfolio has the lowest expected delay. The portfolio model becomes

$$
\begin{aligned}
\text{minimize}\quad & 90z_1{+}67.5z_2{+}90z_3{+}180z_4{+}45z_5{+}90z_6{+}90z_7{+}135z_8{+}180z_{10} \qquad (12)\\
\text{subject to}\quad & \sum_{i=1}^{11} z_i = 1\\
& z_1 + z_2 + z_3 + z_4 \leq C_{33}\\
& z_5 + z_6 + z_7 + z_8 + z_9 + z_{10} \leq C_{57}\\
& z_1 + z_5 + z_6 + z_7 + z_8 \leq C_9\\
& z_2 + z_3 + z_9 \leq C_{41}\\
& z_4 + z_{10} \leq C_{45}\\
& z_{11} \leq C_{66}\\
& z_5 + z_6 + z_3 + z_9 \leq C_{41}\\
& z_7 + z_8 + z_4 + z_{10} \leq C_{45}\\
& z_5 + z_7 + z_3 + z_9 \leq C_{41}\\
& z_1 + z_2 + z_3 + z_6 + z_8 + z_4 + z_{10} \leq C_{45}\\
& z_2 + z_1 + z_5 + z_6 + z_7 + z_8 \leq C_9\\
& z_3 + z_4 + z_{10} \leq C_{45}\\
& \mathbf{z} \in \{0, 1\}
\end{aligned}
$$

5.2 Branch and Price

The Portfolio model does not automatically have integer solutions, so we need to use the method of branch and bound to solve it. Because the number of portfolios is exponentially large, we cannot necessarily solve the linear relaxation with all the columns included at each node in the tree, so we use delayed column generation as well. We first explain using column generation for our problem, then in sections 5.2.1

and 5.2.2, we discuss two specific methods for choosing what column or columns to add. Last, in sections 5.2.3 and 5.2.4, we discuss how to branch strategically for our portfolio model and give an example.

To carry out column generation, we need to be able to calculate the reduced cost of adding in any non-basic variable z_{pq} for $p \in P$, $q \in Q_p$. Each column in our constraint matrix corresponds to a portfolio $q \in Q_p$ for some passenger $p \in P$. Letting $\mathbf{B}$ be the constraint matrix for the current basis, $\mathbf{c}'_B$ be the vector of costs for the basic variables, and A_{pq} be the column of the constraint matrix associated with passenger $p \in P$ and portfolio $q \in Q_p$, the reduced cost for variable z_{pq} is $\bar{c_{pq}} = c_{pq} - \mathbf{c}'_B \mathbf{B}^{-1} \mathbf{A}_{pq}$. All entries in A_{pq} are 0 or 1. We describe which entries are non-zero.

- In the first $|P|$ rows, there is a 1 only in row p.
- In row $|P| + f$, there is a 1 only if $\delta_{qf} = 1$ for $f \in F$.
- In the row corresponding to outcome ω for flight $f \in F$ in the third set of constraints, there is a 1 only if $\delta^{\omega}_{qf} = 1$ or both $\delta_{qf} = 1$ and $\epsilon^{\omega}_{qf} = 0$.

Letting $\mathbf{p}' = \mathbf{c}'_B \mathbf{B}^{-1}$, π_p be the dual variable for the pth constraint in the first set, σ_f be the dual variable for the f'th constraint in the second set, and λ^{ω}_f be the dual variable for the f'th constraint in the third set for outcome ω, then

$$\bar{c_{pq}} = c_{pq} - \left(\pi_p + \sum_{f \in F} \delta_{qf} \sigma_f + \sum_{\omega \in \Omega} \sum_{f \in F} \left(\delta^{\omega}_{qf} + \delta_{qf}(1 - \epsilon^{\omega}_{qf})\right) \lambda^{\omega}_f\right) \tag{5.6}$$

Note that since the elements of $\mathbf{c}_B$ represent delay lengths and all the elements of $\mathbf{B}$ are 1 or 0, the expression in the largest set of parentheses is non-negative.

This reduced cost is not too hard to calculate for each non-basic variable, since we can use complementary slackness to calculate the values of the dual variables. In order to identity the portfolio with the smallest (most negative) reduced cost to add to an individual LP relaxation at a node in the branch and bound tree, we use either the enumeration approach described in section 5.2.2 or an optimization-based pricing problem described in section 5.2.1. In reality, we do not have to add the variables with the most negative reduced cost, but we could add any variables with negative reduced cost. Most likely, adding the variables with most negative reduced cost will speed up run time, since we do not have to add as many columns in total that way, but adding any negative reduced cost variable at each step in column generation will lead us to a minimum cost solution.

While solving each individual LP relaxation at a node of the branch and bound tree, note that we can add more than one variable in each iteration of column generation. In fact, that may prove to decrease the run time. We can add as many variables as we want at a time that have negative reduced cost. We try to balance decreasing the number of iterations that must be completed and keeping the size of the LP relaxation at a node as small as possible. For example, if we add more than one column at a time, then we may end up adding more columns in total by the end of the algorithm at each node than if we added one at a time and then the model is harder to solve at each iteration. On the other hand, if we add more at a time, then we may decrease the total number of iterations that need to be completed at each node.

Next, we can actually enumerate all the reduced costs, then choose the variable or variables with a negative reduced cost. We may have too big of a network and the number of possible columns to add may be big, so that enumerating them all and calculating their reduced cost may be unrealistic in some cases. We can instead solve a separate minimization subproblem to choose what column or columns to add to the model. Each of these options, strategically enumerating the options and solving a subproblem, are explored in the following sections. Note that both methods are generalizable for more than two delayed flights.

5.2.1 Network for Subproblem

In the method of solving a pricing problem to choose columns to add, we need a separate model for each passenger. The only exception is that if two or more passengers have the same original arrival time, destination, and status, then these passengers can share a network. We can solve all $|P|$ of these models and add $|P|$ columns, or we can solve only some of them and add less than $|P|$ columns. Note that using this method, we find the portfolios with the most negative reduced cost, as opposed to one with any negative reduced cost.

The network for passenger p, who was originally scheduled on one (or both) of the delayed flights, represents the passenger's possible portfolios. There are two parts to the network. The first-stage part, representing the itinerary chosen as soon as a delay is discovered, has two types of nodes.

- A start node with supply equal to $|\Omega|$.
- A node with supply 0 for each itinerary that is an option at the current time.

For the first stage network, we make a rule that the flow on each arc must be either 0 or $|\Omega|$, so that a passenger gets exactly one itinerary. We enforce this by having two types of variables, where some of them must be binary, which is shown more clearly in the formulation below.

Next, we have the second-stage part of the network with three types of nodes.

- Scenario nodes. Each itinerary node from the first stage connects to $|\Omega|$ nodes, each one representing a scenario. Different itineraries each have their own set of scenario nodes except in one situation. If two itineraries in scenario $\omega \in \Omega$ put the passenger in the same location at the same time, then these two itineraries share a scenario node for scenario ω. Scenario nodes connect either to flight nodes, described next, if the passenger is misconnected in that scenario, or they connect to the end node.

- Flight nodes. Each scenario has its own network of flight nodes with appropriate connections between them. A flight node connects to the end node if the flight ends in the passenger's desired destination.

- An end node with a demand of $|\Omega|$.

In figure 5.2, we have a picture of the network. In this example, there are two possible first-stage itineraries for the passenger and three different scenarios in Ω. Note that if the passenger receives itinerary i_1 in the first stage, then under scenario ω_1, there is no misconnection and no other flights are needed. After the picture, we have a description of the sets, parameters, variables, and the formulation for the subproblem for each passenger.

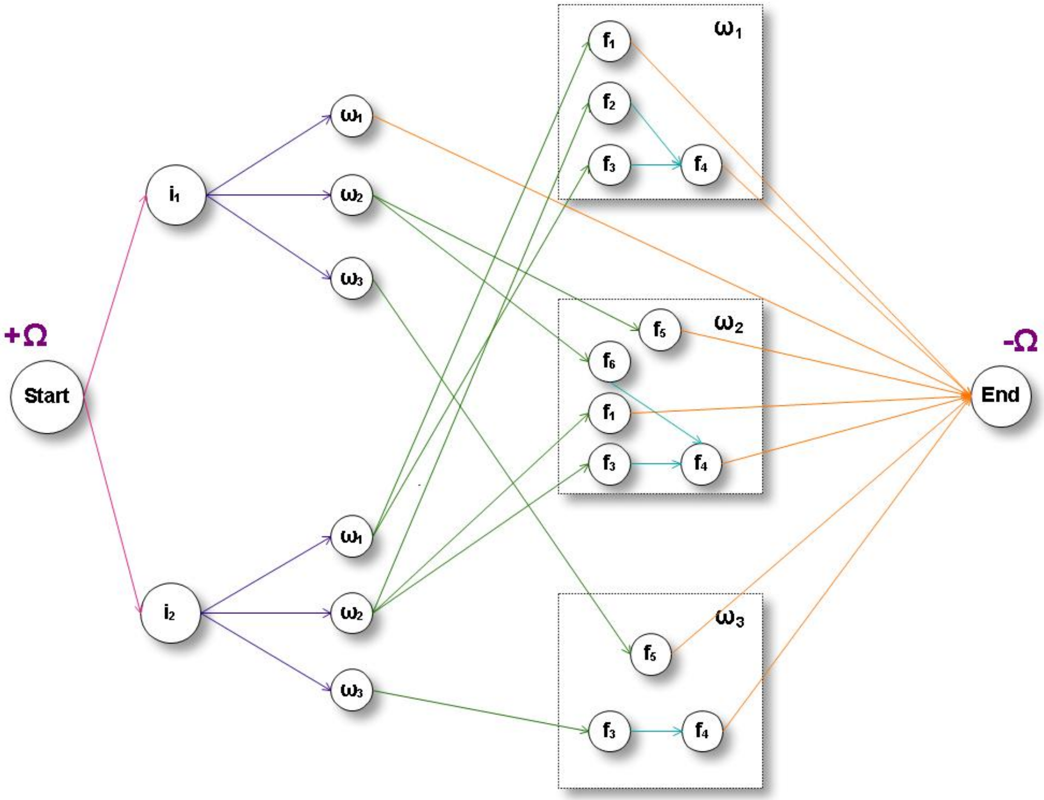

Figure 5.2: Network for Subproblem

Let N be the set of nodes.

Let A be the set of arcs.

Let $A_1 \in A$ be the arcs beginning at the start node.

Let $A_e \in A$ be the arcs going to the end node.

Let $A_e^\omega \in A_e$ be the arcs that begin at a scenario node or one of the second-stage flight nodes for outcome $\omega \in \Omega$ and end at the end node.

Let $A_{ft}^\omega \in A$ be the arcs that end at a second-stage flight node for outcome $\omega \in \Omega$.

Let $B_n \in A$ be the arcs that begin at node $n \in N$.

Let $E_n \in A$ be the arcs that end at node $n \in N$.

Let n_{st} be the start node.

Let n_e be the end node.

Let d_a be the cost of one unit of flow on arc $a \in A$.

Let w_a be the number of minutes of delay the passenger experiences taking arc $a \in A_e$.

Let $p \in P$ be the passenger for whom the network is used.

Let S be the number of scenarios, so $S = |\Omega|$.

Let f_a be the variable for the flow on arc $a \in A$.

Let x_a be a variable that is 1 if arc $a \in A_1$ is used, and 0 otherwise.

$$\text{minimize} \sum_{a \in A} d_a f_a \tag{5.7}$$

$$\text{subject to} \sum_{a \in A_1} f_a = S \tag{5.8}$$

$$\sum_{a \in A_e} f_a = S \tag{5.9}$$

$$\sum_{a \in B_n} f_a - \sum_{a \in E_n} f_a = 0 \qquad \forall n \in N \setminus (\{n_{st}, n_e\}) \tag{5.10}$$

$$f_a = S x_a \qquad \forall a \in A_1 \tag{5.11}$$

$$x_a \leq 1 \qquad \forall a \in A_1 \tag{5.12}$$

$$f_a \geq 0 \qquad \forall a \in A \tag{5.13}$$

$$\mathbf{x} \in \mathbb{Z} \tag{5.14}$$

This is a general minimum cost flow model, except for the fourth, fifth, and seventh sets of constraints. Now we discuss how d_a for each arc $a \in A$ is calculated. Recall that the reduced cost of variable z_{pq} for portfolio $q \in Q_p$ for passenger $p \in P$ is $\bar{c}_{pq} = c_{pq} - \mathbf{c}'_B \mathbf{B}^{-1} \mathbf{A}_{pq}$, where $\mathbf{A}_{pq}$ is the column of the constraint matrix associated with portfolio $q \in Q_p$ for $p \in P$ and $\mathbf{c}_B$ and $\mathbf{B}^{-1}$ are the parts of the cost vector and the constraint matrix for the current basic variables, respectively, for the portfolio model. We want to minimize this quantity by solving the subproblem model. Note that c_{pq} is the expected cost of portfolio $q \in Q_p$ for passenger $p \in P$. Then the first term in the reduced cost is $c_{pq} = \sum_{\omega \in \Omega} \rho^\omega \sum_{a \in A_e^\omega} w_a f_a$.

Let $\mathbf{v}^1$ be the first $|P|$ elements of the vector $\mathbf{c}'_B \mathbf{B}^{-1}$, let $\mathbf{v}^2$ be the next $|F|$ elements of $\mathbf{c}'_B \mathbf{B}^{-1}$, and let $\mathbf{v}^\omega$ be the $|F|$ elements of $\mathbf{c}'_B \mathbf{B}^{-1}$ for the third set of constraints in the portfolio model for $\omega \in \Omega$. Then $\mathbf{c}'_B \mathbf{B}^{-1} \mathbf{A}_{pq}$ equals $v_p^1 + \sum_{f=1}^{|F|} v_f^2 \delta_{qf} + \sum_{\omega \in \Omega} \sum_{f=1}^{|F|} v_f^\omega \big(\delta_{qf}^\omega + \delta_{qf}(1 - \epsilon_{qf}^\omega) \big)$. Since v_p^1 is a constant, we can ignore the first term. Thus, for using flight $f \in F$ in the first stage, we want to assign a cost of $-\big(v_f^2 + \sum_{\omega\in\Omega} v_f^\omega(1-\epsilon_{qf}^\omega)\big)$, and for using a flight $f \in F$ in the second stage in outcome $\omega \in \Omega$, we want to assign a cost of $-v_f^\omega$. For the first-stage flights, we assign the cost to using the arcs in A_1. For example, for itinerary i in the first stage with node n_i, the arc from n_{st} to n_i has cost $\sum_{f\in F_i} -\Big(v_f^2 + \sum_{\omega\in\Omega} v_f^\omega(1-\epsilon_{qf}^\omega)\Big)$, where F_i is the set of flights in itinerary i. For second-stage flights, we assign the cost to all arcs going to the corresponding flight nodes. Note that we know ϵ_{qf}^ω while on an arc in A_1, since the whole itinerary with all its information is considered together. Thus the objective function of our model becomes

$$\sum_{\omega\in\Omega}\rho^\omega \sum_{a\in A_e^\omega} w_a f_a - \left(\sum_{a\in A_1}\Big(v_f^2 + \sum_{\omega\in\Omega} v_f^\omega(1-\epsilon_{qf}^\omega)\Big) f_a + \sum_{\omega\in\Omega}\sum_{a\in A_{ft}^\omega} v_f^\omega f_a\right) \tag{5.15}$$

Note that the reason we use column generation is to speed up the run time and make the problem solvable for a large data set. Still, in some situations it may not speed up the run time and could actually increase it. One situation is when a model is small enough that solving the whole thing with all the columns can be done in a reasonable amount of time. If we instead use column generation, then several iterations may be completed, while each one takes almost as long as solving the original problem just once. In addition, it is not clear what the best method is to use in deciding how many columns to begin with and how many to add at a time, as discussed. Thus, one way of using column generation may produce a quicker solve time, while another method may not.

Last, note that we may have the same problem while solving this network problem that we described in section 5.1.2, where if one flight lands before the delay of the other becomes known, then what scenario is occurring may be unclear. We described how we can create a rule that the second-stage solution for the passengers from one flight must be the same in certain scenarios. Here, for each first-stage itinerary, our network has a separate node for each scenario, so the solutions in different scenarios are always allowed to be different from each other, while we may need the scenario solutions to be the same for the passenger. In order to implement this rule, we can collapse two or more nodes into just one node when necessary. For example, suppose in outcomes ω_1 and ω_2, we need the solution to be the same after a certain first-stage itinerary where the person is disrupted in either case. We then make the nodes for ω_1 and ω_2 into just one node for both scenarios. The arcs coming from the new scenario node are only the ones available for scenario ω_2, where we assume the delay is longer, and they connect to flights in the ω_2 block.

5.2.2 Enumeration for Subproblem

As mentioned, we can enumerate the reduced costs of all the unused columns and pick the one(s) with the lowest reduced costs to add to the model. As described above, the reduced cost for variable z_{pq} associated with portfolio $q \in Q_p$ for passenger $p \in P$ is $\bar{c_{pq}} = c_{pq} - \mathbf{c}'_B \mathbf{B}^{-1} \mathbf{A}_{pq}$. This is the same as

$$c_{pq} - \left(\pi_p + \sum_{f \in F} \delta_{qf} \sigma_f + \sum_{\omega \in \Omega} \sum_{f \in F} \left(\delta^{\omega}_{qf} + \delta_{qf}(1 - \epsilon^{\omega}_{qf}) \right) \lambda^{\omega}_f \right) \tag{5.16}$$

As in the previous section, we consider each passenger separately. For each passenger $p \in P$, we enumerate the reduced cost for variables z_{pq} for all $q \in Q_p$. We choose the one with the lowest reduced cost and add it to the set Z^{best}. We repeat this process for all $|P|$ passengers and thus end up with $|P|$ columns in the set Z^{best}. We can add all these columns to the model. There are a couple of other options related to this.

- We can do this process for less than all of the passengers and so we add less than $|P|$ columns to the model.
- For each passenger, we do not have to enumerate all the possible portfolios, but we could instead enumerate them just until we find one with negative reduced

cost, and add that column to Z^{best}.

- For each passenger, we can add more than one column with negative reduced cost to Z^{best} to add to the model.

We are interested in the run time of enumerating all the portfolios. A mathematical expression for the number of portfolios is

$$\sum_{p \in P} \sum_{i \in I_p} \prod_{\omega \in \Omega} |I_p^\omega| \tag{5.17}$$

where Ω is the set of possibilities for both delayed flights and I_p^ω is the set of possible second-stage itineraries if the passenger $p \in P$ has a misconnection in scenario $\omega \in \Omega$. Since not all itineraries $i \in I_p$ will cause a disruption in a certain scenario $\omega \in \Omega$ for passenger $p \in P$, let $\Omega_i^M \subseteq \Omega$ be the set of scenarios where there is a disruption with itinerary i for the passenger. Note that this set can be empty, a proper subset of Ω, or equal to Ω. The number of portfolios in the model can be written as

$$\sum_{p \in P} \sum_{i \in I_p} 1 \prod_{\omega \in \Omega_i^M} |I_p^\omega| \tag{5.18}$$

Note that the number of second-stage itineraries after a misconnection depends on the scenario $\omega \in \Omega$, since with shorter delays, more flights are still available. Note also that in a certain scenario $\omega \in \Omega$, the number of second-stage itineraries depends on what delayed flight is part of the first-stage itinerary $i \in I_p$ given to the passenger $p \in P$, assuming that i caused a misconnection, but the number does not depend on what other flights are part of itinerary i. Last, a candidate for a second-stage itinerary obviously must have enough time between the delayed flight and the next flight in the scenario that occurs so that the passenger can make it to the second flight. These second-stage itineraries can have up to 3 flights.

To study the run time for enumerating the portfolios, we investigate how many portfolios there are for each passenger. We assume there is only one delayed flight for simplicity, although there are two delayed flights in this model. We find that in one simulation of instance 1, 3, and 5, the number of portfolios is 16, 3799, and $4,021,612,168$, respectively. Computing the reduced cost for this many portfolios is probably feasible in a short time, since calculating the number of portfolios takes only a second or so. If it turns out that computing the reduced cost this many times takes too long, then we could use one of the options above. Each of them has the possibility of speeding up the total run time. Enumerating columns for only some of the passengers will give a better result, but not all the reduced costs have to be calculated. Also, choosing the first negative reduced cost column found for each passenger could be much quicker, and adding several negative reduced columns at once for each passenger means we will not have to enumerate the reduced costs as many times.

Since we have simulated the number of portfolios for only one delayed flight and in only three instances, we now investigate the order of the number of portfolios. For simplicity, we let the number of passengers be P, the number of flights be F, and the number of scenarios be Ω. We find that the number of portfolios is $O\big(PF^{3(1+\Omega)}\big)$. To see this, the order of the number of first-stage itineraries for each passenger is F^3 since itineraries can have up to 3 flights. Next, for each scenario that an itinerary is disrupted, the order of the number of second-stage itineraries is F^3. Thus the order of the number of portfolios corresponding to one specific first-stage itinerary $i \in I_p$ is $F^{3\Omega}$. Multiplying, the total number of portfolios is $O(PF^3F^{3\Omega}) = O(PF^{3(1+\Omega)})$.

5.2.3 Branching

After column generation has been completed and the optimal solution has been found at the root node of the branch and bound tree (the LP relaxation), the solution may be fractional. Thus, we need to branch in order to find an integral solution. The situation of using both branch and bound and column generation is often called branch and price, which is described very well in the paper *Barnhart et al.* (1998). Note that we carry out column generation at each of the individual nodes of the branch and bound tree when necessary, so column generation happens within branch and bound.

When performing branch and bound, if a fractional solution is encountered, then the problem is split into two separate models, each of which is solved to optimality. Normally while using branch and bound, a constraint is added to each of these problems. For example, if the solution at the root node includes $z_{5,1} = \frac{1}{2}$, then the constraint $z_{5,1} = 0$ is added to one of the problems, and $z_{5,1} = 1$ is added to the other in order to force $z_{5,1}$ not to be fractional again.

Unfortunately, since there is now another constraint in each model, the structure of each model has changed. There is a new dual variable for each, which makes the expression for the reduced cost no longer valid. Since the reduced cost is $\bar{c_{pq}} = c_{pq} - \mathbf{p}'\mathbf{A}_{pq}$, where $\mathbf{p}$ is the dual variables, another term is added to the expression 5.6 above for each constraint added to a model.

Continuing the example, suppose that the solution at the root node includes $z_{5,1} = \frac{1}{2}$, so we add the constraint $z_{5,1} = 0$ to one of the problems, and the constraint $z_{5,1} = 1$ to the other problem in order to force $z_{5,1}$ not to be fractional again. We let $\gamma^0_{5,1}$ and $\gamma^1_{5,1}$ be the dual variables for these constraints. We add either $\gamma^0_{5,1}$ or $\gamma^1_{5,1}$ to the expression for the reduced cost, $c_{pq} - \left(\pi_p + \sum_{f\in F} \delta_{qf}\sigma_f + \sum_{\omega\in\Omega}\sum_{f\in F}\left(\delta^{\omega}_{qf} + \delta_{qf}(1-\epsilon^{\omega}_{qf})\right)\lambda^{\omega}_f\right)$. As more and more nodes are reached in the tree, then more γ terms must be added to the expression for the reduced cost, making the computations more difficult. Thus, we want to devise a different way to enforce rules such as $z_{5,1} = 0$ and $z_{5,1} = 1$.

We present another method to enforce a cut while forming two new nodes in the branch and bound tree without adding any new constraints. Some work on using branch and price and performing branching strategies are described in the papers

Appelgren (1969), *Johnson* (1989), *Ryan and Foster* (1981), and *Barnhart et al.* (1998). If we have a constraint making it so that a certain portfolio variable is 0, this means that this portfolio is never used again in this part of the tree. Instead of adding the constraint, we can remove the column for that variable from the model. At that node, we solve the model again without that column. For the node where we set the variable equal to 1, this effectively says that the corresponding portfolio is the only choice for the passenger in that part of the tree. We can remove all other columns for variables for that passenger from the model. At this node, we solve the model with these columns removed. How does this affect the expression for the reduced cost of non-basic variables? The answer is that it does not at all. Since that expression only depends on each column's entries, removing one or more columns does not change the others.

In the example, suppose that at the root node, part of the solution is that $z_{5,1} = \frac{1}{2}$. At the node where we choose not to use portfolio 1 for passenger 5, we remove the column for variable $z_{5,1}$ from the model. At the node where we choose to use only portfolio 1 for passenger 5, then we remove the columns for variables $z_{5,2}, ..., z_{5,|Q_p|}$ and solve the model with these columns removed.

Note that we purposely remove one or several columns for a specific part of the tree. Thus, we need to ensure that the variables do not enter into the basis again as we carry out column generation in that part of the tree, either at the current node or at future nodes. Variables that are removed while branching may often have negative reduced costs, since they were chosen as good solutions previously, so may be suggested during column generation. Causing columns that were purposely removed not to be added back into the basis is much more simple for the enumeration method described in section 5.2.2 than it is for the network method for the subproblem described in section 5.2.1. If we use enumeration for the subproblem, we can keep a list for each node showing which columns are not eligible to be added to the model. If while calculating reduced costs to look for a column or columns to add, one of these ineligible columns is encountered, then we throw it out and choose the next best one. If one or more ineligible columns are the only ones with negative reduced cost at a certain node, then the process of column generation is complete at that node. On the other hand, if we use the network method for the subproblem and the solution we find is a column that is ineligible, then we cannot so easily throw it out and pick the next best one. We suggest this as a topic for future research.

5.2.4 Example of Solving Methods

We illustrate the different solving methods by continuing the example from section 5.1.4. Recall that there are two delayed flights and one passenger p. The passenger's original itinerary is Flight 33 from PDX to LAX, leaving at 8:20 am and landing at 11:08 am, and Flight 41 to location ATL leaving at 2:00 pm and arriving at 7:00 pm. There is a delay of either one hour or three hours on flight 33, so the passenger may miss her connection. The second delayed flight is Flight 9, with a delay length of 30 minutes or 2.5 hours. All flights that are relevant to the passenger, all portfolios made up of these flights, and the model are described in section 5.1.4.

By inspection of the model, note that one solution is $z_{p,9} = 1$ and another is $z_{p,11} = 1$, where all other variables are 0. Since we cannot usually find the solution by inspection, we illustrate our method of solving the model. We start with a feasible solution by using the column corresponding to portfolio 1. Considering the linear relaxation, we solve the model. Then using column generation and letting all available capacities be 1, we add in the columns for z_9 and z_{11} to form the model

$$\begin{aligned} \text{minimize} \quad & 90z_1 \\ \text{subject to} \quad & z_1 + z_9 + z_{11} = 1 \\ & z_1 \leq 1 \\ & z_9 \leq 1 \\ & z_{11} \leq 1 \\ & z_1, z_9, z_{11} \geq 0 \end{aligned} \tag{13}$$

In order to illustrate branching, suppose that we have the solution $z_1 = 0$, $z_9 = \frac{1}{2}$, $z_{11} = \frac{1}{2}$ with value 0 (even though this is not a basic feasible solution). Since all variables not in model (13) have non-negative reduced cost, we have reached a solution to the original problem (1). Since the solution is not integral, we need to branch. For one of the new problems, we want to enforce the constraint $z_9 = 0$, and in the other $z_9 = 1$, so that z_9 will never be fractional again. We can follow two different methods.

- Add the constraints to the two models. The two models become

  $$\begin{aligned} \text{minimize} \quad & 90z_1 \\ \text{subject to} \quad & z_1 + z_9 + z_{11} = 1 \\ & z_1 \leq 1 \\ & z_9 \leq 1 \\ & z_{11} \leq 1 \\ & z_9 = 0 \\ & z_1, z_9, z_{11} \geq 0 \end{aligned}$$

  $$\begin{aligned} \text{minimize} \quad & 90z_1 \\ \text{subject to} \quad & z_1 + z_9 + z_{11} = 1 \\ & z_1 \leq 1 \\ & z_9 \leq 1 \\ & z_{11} \leq 1 \end{aligned}$$

$$z_9 = 1$$

$$z_1, z_9, z_{11} \geq 0$$

- Remove columns. For the first problem, we remove the column for z_9 and in the second problem, we remove the columns for z_1 and z_{11}. Thus, our two problems become

$$\begin{aligned} \text{minimize} \quad & 90z_1 \\ \text{subject to} \quad & z_1 + z_{11} = 1 \\ & z_1 \leq 1 \\ & z_{11} \leq 1 \\ & z_1, z_{11} \geq 0 \end{aligned}$$

$$\begin{aligned} \text{minimize} \quad & 0 \\ \text{subject to} \quad & z_9 = 1 \\ & z_9 \leq 1 \\ & z_9 \geq 0 \end{aligned}$$

Using either method, basic feasible solutions to these two problems are $(z_1, z_{11}) = (0, 1)$ and $z_9 = 1$, respectively, each of which has cost 0. Now since each solution is integral, we are done branching. The two best integral solutions found in this whole process are giving the passenger portfolio 9 or 11, as we predicted at the beginning. Note that since we do not need to add columns to our two new models and thus find the reduced cost of other variables, the benefit of using the second method for branching is not as clear.

5.3 Special Cases

Since the portfolio model may have many variables, each of which is binary, we have to employ several methods to decrease the solve time, as described in the previous sections. There are two special cases where the problem becomes much simpler and we can make use of this simplicity to solve the problem in a different way. First, suppose that for the two delayed flights, no passenger could be assigned both flights and no two passengers on each of those flights would ever be assigned another flight in common. This may occur if the flights have certain origins and destinations, planned times of take-off, and certain chances of delay. In that case, we can solve the problem as two separate problems, where each has only one delayed flight. We can model each problem as in section 3.2.6 and solve using Benders decomposition as in chapter IV.

Second, suppose that all of the flights in the data set that are in the list of possible itineraries for at least one passenger have capacities that are no less than the number of passengers that could use those flights, which is uncommon in practice. In this situation, we can solve a separate problem for each passenger, since there are no constraints that will be active that involve more than one person. This should speed up solve time substantially, since each model then is relatively small. We do still have to enforce integrality constraints if we continue to use the portfolio model for each passenger in case more than one possible portfolio has the same lowest cost, so that a solution with fractional values could be chosen. Then we would have to carry out branch and bound. If no possible portfolios have the same lowest cost, then an integral solution will automatically be chosen. Alternatively, we could make the decisions for all the passengers in the way we did in section 3.2.6, since the network for the second-stage problem only needs to have one start node for one passenger. Solving this may be faster than solving the portfolio model, since in the combined model, the solution is automatically integral for the second-stage problem.

5.4 More Than Two Flights Delayed

Suppose now that an airline is dealing with many delayed flights around the same time. In particular, there could be delayed flights in different cities, even all over an airline's network, or the delayed flights could all be at the same airport. Also, the flights could be related in many different ways, such as being possible connections from each other, being substitutes for each other, sharing the same aircraft, etc. The number of possible scenarios considering all the different delay possibilities could be very high, and trying to make good decisions for all affected passengers based on so many unknowns can be extremely difficult.

Fortunately, the portfolio model for two delayed flights can be easily extended to deal with this type of situation where there are many delayed flights. When the portfolio model is extended for any number of delayed flights, Ω is the set of possible outcomes for the delays of all delayed flights considered, and P is the set of passengers on all those delayed flights. Letting Ω_n be the set of possible delays for the n'th delayed flight and letting N be the number of delayed flights, then the size of Ω could be up to $|\Omega_1||\Omega_2|\cdots|\Omega_N|$, which can be quite large.

A good method for solving the portfolio model when there are many delayed flights is the one described in section 5.2 for when there are just two delayed flights. We can still use branch and price, have the option of enumeration or a network to find what column(s) to add during column generation, and can add constraints or remove columns for different nodes in the branch and bound tree.

Since the model could be huge, we are not sure how quickly the model can be solved as more flights and possibilities for delay lengths are added to the model. We may have to use other methods in addition in order to speed up the run time, such as heuristics or delayed constraint generation. If the model is too large to solve at one time, we could break it down into several smaller problems, grouping together sets of delayed flights that have the most in common in terms of possible shared passengers or time of day. We could then solve them sequentially. This may not necessarily produce the optimal solution, but it would speed up the solve time and produce a good, feasible solution. The solution for one set of flights would have to be used as an input for the next problem, since available capacities are changed. We could also look into using Benders decomposition, where the master problem represents one set of delayed flights and subproblems are for other delayed flights. It is possible that some delayed flights' problems are not even related, because of timing and location, so that solving their problems together would not even be beneficial.

Note that a passenger could be assigned several delayed flights in his or her chosen portfolio. For example, there could be two delayed flights in the first-stage itinerary, then another delayed flight in the second-stage itinerary for some scenarios. There is still the rule that a passenger cannot be assigned a delayed flight in the second stage if it may lead to a misconnection, because we do not have a third-stage problem in this model. That situation could be considered in the future.

Thus, even though we solve the model as a two-stage process, we may execute it in a multi-stage process as the different flights land and have disrupted passengers at different times of the day. The number of stages in which we execute the solution may change based on the number of delayed flights and their details. Note that when there are $n > 2$ delayed flights, we may have to execute the solution in $n + 1$ stages.

CHAPTER VI

Conclusion and Future Research

In this dissertation, we proposed new methods to improve the reaccommodation of airline passengers after delays and cancellations of flights. We proposed several models for a deterministic delay length and several two-stage stochastic programming problems for when the delay length is not known for certain. Also, we have written a model and devised a solving method for the situation with two or more delayed flights.

In chapter III, one significant contribution was a two-stage stochastic model where the first-stage variables represent itineraries and the second-stage variables represent flow on arcs in a network for the flights. In the future, we would like to consider a few variations on this model and the others in chapter III. First, we want to consider different objective functions from what we currently have, which is the sum of the delays for all passengers. It is possible that an objective function that more accurately represents priorities in reality would consider the worst delay experienced by any one passenger, the difference between different passengers' delays, the number of flights in chosen itineraries, and other factors. Another interesting variation on the model is allowing the decision to be made to hold a connecting flight for a passenger who is on a delayed flight. In adding this possibility to our model, we would set the rule that any connecting flights could be delayed by only a certain amount so that no passengers on that flight would then miss later connections. This option could improve the solution in cases where a passenger's only other option for a connecting flight is much later in the day. We would also like to consider letting first-stage itineraries have only one or two flights instead of three. This rule would promote equity between passenger so that one passenger does not get a bad itinerary so that several other passengers can have better ones.

In chapter IV, we implemented the combined model in section 3.2.6, solving it as a MIP problem and also using Benders decomposition. We compared the results for the delay experienced by passengers using our model and using the method of reaccommodating passengers one-by-one after misconnections have occurred. Next, we want to consider solving the model with a bigger data set. For example, we could use flights from more than one day while taking into account the cost to the airline of providing hotel rooms and meals for passengers who are delayed overnight.

There are also several ways we would like to try solving our model faster. One is to streamline which itineraries are options for the passengers in the first-stage problem. Currently, there may be itineraries in a passenger's list that would never be chosen because they occur too late in the day based on the possible lengths of delay. Also, we could possibly eliminate multi-leg itineraries that do not connect at hubs since they are probably not cost-effective. This could make the problem smaller so it can be solved faster. Next, we would like to investigate how we could consolidate scenarios and passengers that are alike in order to make setting up and solving the model faster. It would be interesting to look at other ways to analyze the data in the future, such as using the median instead of the average, using the worse cost for a scenario or passenger, etc.

In reality, there is often not only one delayed flight, but many different delays and cancellations all affecting each other during daily operations for an airline. In chapter V, we modeled the situation with two or more delayed flights. We proposed some methods of solving it involving branch and price, which includes branch and bound and column generation, and strategic enforcement of constraints while branching. Next, we are very interested in implementing the portfolio model and analyzing the results. Since all the variables are binary, implementing this may involve more methods of decreasing the run time.

One other topic of interest is the possibility of airlines cooperating in reaccommodating disrupted passengers. We want to work on how to include in our models the option of moving passengers to another airline's flights. This is basically the same problem as before except that there is a much larger set of flights and the cost function is more complicated. The cost for this situation would involve not only the delay for the passengers but also payment to the other airline. It would be interesting to model this cost compared to the cost of delay for passengers. In many situations, this extra cost may be worth it in order to get passengers to their destinations more quickly.

In addition, we are interested in exploring the topics mentioned in section 3.2.8 where our research may be relevant to other problems where one or more pieces of information in a network is unknown.

APPENDICES

APPENDIX A

Table of Notation

Below are tables with descriptions for all of the sets, parameters, and variables used in sections 3.1, 3.2, and 5.1.3. The third column shows which models apply for each description. The letters I, F, M, C, and P correspond to the itinerary-based (3.1.1 and 3.2.3), flight-based (3.1.2 and 3.2.4), minimum cost flow (3.1.3 and 3.2.5), combined (3.2.6), and portfolio (5.1.3) formulations.

Sets	**Description**	**Models**
P	Passengers on the original flight	I, F, M, C, P
F	Flights considered	F, M, C
I_p	Possible itineraries for passenger $p \in P$ in the first stage	I, C
F_i	Flights from itinerary $i \in I_p$ for passenger $p \in P$	I, C
N	All nodes	F, M, C
N_d	Nodes for the destinations	M
N_p	End nodes for the passengers	C
N_{fd}	Nodes that are first in a pair for flights that begin at the dest. of the delayed flight	F
A	All arcs	F, M, C
A_c^{s1}	Arcs in 1st-stage network that begin at the 2nd flight node of the delayed flight and connect to another flight node	F, M
A_e	Arcs ending at an end node	F, M, C
A_{fp}	Arcs between two flight nodes in a pair	F, M, C
A_{fpi}	Arcs in A_{fp} in the flight-based model that correspond to itinerary i in the itinerary-based model	C
B_n	Arcs that begin at node $n \in N$	F, M, C
E_n	Arcs that end at node $n \in N$	F, M, C
Ω	Possible outcomes for the length of delay of the delayed flight(s)	I, F, M, C, P
I_p^ω	Possible itineraries for passenger $p \in P$ in stage two in outcome $\omega \in \Omega$	I
A^ω	Arcs in the second stage in outcome $\omega \in \Omega$	F, M, C

Sets	Description	Models
$B_{n_1}^{\omega}$	Arcs that begin at the start node in the second stage in outcome $\omega \in \Omega$	F, M, C
E_n^{ω}	Arcs that end at node $n \in N$ in the second stage in outcome $\omega \in \Omega$	R, M, C
Q_p	Possible portfolios for passenger $p \in P$	P

Parameters	Description	Models
C_f	Available capacity for flight $f \in F$	I, C
c_{pi}	Delay cost for passenger $p \in P$ with itinerary $i \in I_p$	I, C
n_1	Start node	F, M, C
n_l	End node	F, C
n_p	End node for passenger $p \in P$	C
s_n^p	Supply of commodity $p \in P$ at node $n \in N$	F
s_n	Supply at node $n \in N$	F, M
u_a^p	Capacity of arc $a \in A_e$ for passenger $p \in P$	F, C
u_a	Total capacity of arc $a \in A_{fp}$	F, M, C
c_a^p	Delay cost for passenger $p \in P$ on arc $a \in A_e$	F, C
c_a	Cost for a passenger on arc $a \in A_e$	M
ρ^{ω}	Probability of outcome $\omega \in \Omega$	I, F, M, C
cap	Pre-defined constant	I, F, M, C
dif	Pre-defined constant	I, F, M, C
c_{pq}	Expected cost of portfolio $q \in Q_P$ for passenger $p \in P$	P

Variables	**Description**	**Models**
x_{pi}	Binary decision variable, 1 if passenger $p \in P$ gets itinerary $i \in I_p$	I, C
x_a^p	Binary decision variable, 1 if arc $a \in A$ is used for passenger $p \in P$	F
x_a	Decision variable, number of passengers on arc $a \in A$	M
y_{pi}^{ω}	Binary decision variable, 1 if passenger $p \in P$ gets itinerary $i \in I_p^{\omega}$ in outcome $\omega \in \Omega$ in the second stage	I
$y_a^{p\omega}$	Binary decision variable, 1 if passenger $p \in P$ uses arc $a \in A$ in stage two in outcome $\omega \in \Omega$	F, C
y_a^{ω}	Decision variable, flow on arc $a \in A$ in the second stage with outcome $\omega \in \Omega$	M
z_{pq}	Binary decision variable, 1 if passenger $p \in P$ gets portfolio $q \in Q_p$	P
ϵ_i^{ω}	Binary, 1 if itinerary i is disrupted in outcome $\omega \in \Omega$	I, C
ϵ_a^{ω}	Binary, 1 if arc $a \in A_c^{s1}$ cannot be used in outcome $\omega \in \Omega$	F, M
ϵ_{qf}^{ω}	Binary, 1 if misconnection with portfolio $q \in Q_p$ causing flight $f \in F$ to be missed in outcome $\omega \in \Omega$	P
δ_{ad}^{ω}	Flow planned for arc $a \in A_c^{s1}$ in stage one to end at destination d disrupted in outcome $\omega \in \Omega$	M
δ_a^{ω}	Number of passengers planned for arc $a \in A_e \cup A_{fp}$ in stage one but disrupted in outcome $\omega \in \Omega$	M

Variables	**Description**	**Models**
δ_{qf}	Binary, 1 if flight $f \in F$ is in 1st-stage part of portfolio $q \in Q_p$	P
δ_{qf}^{ω}	Binary, 1 if flight $f \in F$ is in the 2nd-stage part of portfolio $q \in Q_p$ in outcome $\omega \in \Omega$	P

APPENDIX B

Computational Results

Instance 2

Sim	Method	EAC	EAC r.	END	END30	Cost%	ENNR	EHC
Ave	New	85.7	74.7	25.4	21.4	65%	0	255.2
Ave	Cur.	116.4		29.3	25.3		2.7	2231.2
1	New	92.8	80.2	27.5	23.5	62%	0	232.25
1	Cur.	128.4		31.5	27.5		4.25	2201
2	New	64.3	59.4	22.3	18.3	70%	0	174.5
2	Cur.	84.3		23.8	19.5		1.25	1245.25
3	New	76.5	72.9	24.0	20.0	73%	0	157.5
3	Cur.	99.8		28.3	24.3		1.25	2190
4	New	82.3	78.9	27.8	23.8	76%	0	217
4	Cur.	103.3		26.3	22.3		0.75	1264
5	New	89.6	81.5	26.3	22.3	69%	0	210.75
5	Cur.	118.5		29.3	25.3		2.5	2261.75
6	New	92.1	81.2	24.8	20.8	64%	0	334.5
6	Cur.	127.2		29.5	25.5		2.75	2276.5
7	New	68.9	59.3	22.3	18.3	55%	0	195.25
7	Cur.	108.5		28.3	24.3		2.5	2201
8	New	63.8	56.1	20.0	16.0	62%	0	157.5
8	Cur.	90.5		26.3	22.3		2.25	2190
9	New	92.1	82.8	27.3	23.3	70%	0	208
9	Cur.	117.6		31.3	27.3		3.25	2205.75
10	New	114.5	94.5	27.5	23.5	64%	0	412.5
10	Cur.	148.5		33.8	29.8		3.25	3247.5
11	New	69.4	60.9	22.0	18.0	70%	0	168.5
11	Cur.	87.3		26.0	22.0		2.5	2190
12	New	103.7	78.3	30.3	26.3	53%	0	428.5
12	Cur.	147.2		33.8	29.8		4.75	3247.5
13	New	100.9	81.5	25.3	21.3	57%	0	334.5
13	Cur.	144.2		32.0	28.0		4	3247.5
14	New	68.6	63.5	22.3	18.3	62%	0	157.5
14	Cur.	102.3		27.5	23.5		1.5	1197.5
15	New	106.6	89.1	31.3	27.3	64%	0	438.5
15	Cur.	138.7		32.3	27.8		3.25	2302.5

Instance 3

Sim	Method	EAC	EAC r.	END	END30	Cost%	ENNR	EHC
Ave	New	96.5	72.1	51.8	43.6	63.4%	0.05	513.6
Ave	Cur.	114.0		57.3	49.1		10.6	2706.7
1	New	105.9	82.2	55.0	46.8	63%	0	412.5
1	Cur.	131.4		61.5	53.3		10.5	3247.5
2	New	105.5	77.3	58.8	50.5	68%	0	438.5
2	Cur.	113.0		57.5	49.3		10.8	3247.5
3	New	122.4	89.0	61.3	53.0	66%	0	467.75
3	Cur.	135.4		63.8	55.5		14.0	3247.5
4	New	100.6	80.6	55.5	47.3	70%	0	474.25
4	Cur.	115.4		57.0	48.8		9.8	3247.5
5	New	68.9	54.2	40.0	31.8	73%	0	192.5
5	Cur.	74.5		45.3	37.0		7.3	2190
6	New	124.5	85.5	55.0	46.8	62%	0	412.5
6	Cur.	136.9		62.5	54.3		14.5	3247.5
7	New	69.1	52.3	41.5	33.3	51%	0	184.25
7	Cur.	103.1		56.0	47.8		8.8	2231.25
8	New	75.2	68.9	45.5	37.3	65%	0	181.25
8	Cur.	105.8		54.3	46.0		3.8	2261.5
9	New	83.2	64.0	51.8	43.5	61%	0	230
9	Cur.	104.7		54.3	46.0		10.8	2190
10	New	87.4	70.2	51.0	42.8	55%	0	228.25
10	Cur.	128.0		60.5	52.3		9.8	2250.75
11	New	149.8	87.5	58.5	50.3	65%	0.8	3276.5
11	Cur.	135.1		62.3	54.0		12.5	3247.5
12	New	89.5	66.5	51.8	43.5	55%	0	350.5
12	Cur.	121.8		58.5	50.3		11.0	2326.5
13	New	72.6	48.4	42.5	34.3	53%	0	280.5
13	Cur.	91.1		53.3	45.0		11.8	2197
14	New	111.9	92.9	62.3	54.0	73%	0	285
14	Cur.	127.3		62.3	54.0		13.3	3247.5
15	New	81.7	62.2	47.3	39.0	71%	0	290.25
15	Cur.	87.1		51.3	43.0		10.5	2220.5

Instance 4

Sim	Method	EAC	EAC r.	END	END30	Cost%	ENNR	EHC
Ave	New	92.7	80.3	53.6	45.3	67%	0	371.0
Ave	Cur.	119.1		58.4	50.1		5.5	2683.3
1	New	104.2	93.6	60	51.8	68%	0	350.5
1	Cur.	138.2		64.3	56		5.8	2326.5
2	New	94.2	79.6	53.3	45	68%	0	438.5
2	Cur.	116.4		57.8	49.3		5.8	2304.5
3	New	103.3	94.4	59.3	51	70%	0	428.5
3	Cur.	133.9		60.5	52.3		3.3	3247.5
4	New	91.3	75	53	44.8	69%	0	428.5
4	Cur.	108.6		54.5	46.3		6.5	3247.5
5	New	84.2	67	48.3	39.5	65%	0	412.5
5	Cur.	103.3		56.5	48		7.3	3247.5
6	New	92.2	81.6	55.3	47	66%	0	307
6	Cur.	123.2		60.8	52.3		4.8	2301.75
7	New	86.3	81.3	52	43.8	75%	0	236.75
7	Cur.	107.8		55.5	47		3	2307.5
8	New	99.8	79.8	53.25	45	65%	0	438.5
8	Cur.	122.9		60	51.5		7	3247.5
9	New	100.5	87.7	56.3	48	70%	0	438.5
9	Cur.	124.6		60.3	52		4.5	3247.5
10	New	88	78.1	52	43.8	67%	0	360.5
10	Cur.	116.9		56	47.8		4.3	2304.5
11	New	86.9	71.7	48.5	40.3	64%	0	350.5
11	Cur.	111.4		54.3	46		7.3	2312.25
12	New	89.4	74.8	52	43.8	63%	0	348.3
12	Cur.	119.3		59.8	51.5		7.5	2326.5
13	New	96.4	86.7	59.3	50.5	64%	0	298
13	Cur.	135.2		62.8	54.3		5.5	3247.5
14	New	87.8	79.9	52.8	44.5	69%	0	350.5
14	Cur.	116.5		58.8	50.5		4	2276.5
15	New	86.3	74.1	48.8	40.5	69%	0	378
15	Cur.	107.8		54.5	46.3		5.8	2304.5

Instance 5

Sim	Method	EAC	EAC r.	END	END30	Cost%	ENNR	EHC	ELC
Ave	New	159.4	110.2	63.8	61.8	54%	0	390.1	0.0
Ave	Cur.	204.0		72.0	69.8		18.7	3681.7	21.9
1	New	167	114	64	62	51%	0	445	0
1	Cur.	223		75	73		19	4052	0
2	New	117	83	54	52	52%	0	287	0
2	Cur.	161		61	59		13	3303	0
3	New	172	132	68	66	63%	0	380	0
3	Cur.	210		74	72		18	3531	0
4	New	163	118	63	61	50%	0	435	0
4	Cur.	238		79	77		18	3822	0
5	New	195	166	81	79	72%	0	364	0
5	Cur.	232		77	75		16	4052	0
6	New	158	89	62	60	45%	0	420	0
6	Cur.	199		71	69		23	3550	30
7	New	167	117	75	73	57%	0	361	0
7	Cur.	204		73	71		20	3801	58
8	New	127	87	53	51	51%	0	270	0
8	Cur.	172		65	63		15	3259	0
9	New	157	112	61	59	58%	0	391	0
9	Cur.	193		70	68		20	3791	0
10	New	184	95	62	60	40%	0	508	0
10	Cur.	237		80	78		28	4052	0
11	New	164	118	68	66	54%	0	341	0
11	Cur.	220		76	73		20	3550	131
12	New	164	104	65	63	54%	0	510	0
12	Cur.	192		70	67		19	3790	0
13	New	139	101	64	62	55%	0	299	0
13	Cur.	182		67	65		15	3786	0
14	New	173	116	65	63	52%	0	407	0
14	Cur.	222		76	74		22	3309	109
15	New	144	100	53	51	57%	0	434	0
15	Cur.	177		65	63		15	3579	0

Instance 6

Sim	Method	EAC	EAC r.	END	END30	Cost%	ENNR	EHC
Ave	New	157.9	89.9	126.0	121.2	50%	0.1	645.1
Ave	Cur.	178.3		136.3	132.1		49.4	3739.5
1	New	201	94	124	119	55%	1.9	3817
1	Cur.	171		134	130		49	4052
2	New	199	121	152	148	66%	0	513
2	Cur.	182		143	138		59	3786
3	New	194	122	148	144	57%	0	567
3	Cur.	213		153	149		62	4052
4	New	135	58	109	105	34%	0	349
4	Cur.	172		136	132		55	3266
5	New	122	81	108	104	61%	0	362
5	Cur.	132		109	105		28	3525
6	New	148	84	119	114	47%	0	407
6	Cur.	178		135	131		47	3791
7	New	154	81	145	131	42%	0	369
7	Cur.	195		147	142		56	3793
8	New	150	95	128	124	55%	0	389
8	Cur.	173		134	130		43	3789
9	New	168	86	130	126	42%	0	439
9	Cur.	204		150	146		61	4052
10	New	143	89	120	116	51%	0	353
10	Cur.	173		132	128		42	3786
11	New	172	100	136	132	57%	0	508
11	Cur.	176		135	131		51	3786
12	New	111	67	90	86	46%	0	281
12	Cur.	145		117	113		31	3527
13	New	161	102	135	130	56%	0	381
13	Cur.	183		138	134		48	3566
14	New	133	73	102	98	45%	0	430
14	Cur.	163		129	125		45	3536
15	New	177	95	146	141	45%	0	513
15	Cur.	213		154	150		63	3786

Instance 7

Sim	Method	EAC	EAC r.	END	END30	Cost%	ENNR	EHC
Ave	New	123.9	78.6	116.2	111.3	52%	0	330.6
Ave	Cur.	152.2		134.0	128.9		42.3	3758.9
1	New	144	86	127	122	47%	0	491
1	Cur.	182		148	143		50	4105
2	New	136	86	128	124	48%	0	324
2	Cur.	178		151	146		54	4105
3	New	108	65	108	103	45%	0	271
3	Cur.	143		129	124		39	4106
4	New	150	106	130	125	75%	0	505
4	Cur.	142		129	124		38	3790
5	New	101	60	100	95	45%	0	207
5	Cur.	135		124	119		36	3149
6	New	124	78	115	110	53%	0	378
6	Cur.	148		132	127		43	3188
7	New	121	88	123	118	66%	0	229
7	Cur.	132		127	122		39	3149
8	New	152	105	136	131	56%	0	412
8	Cur.	186		147	142		46	4106
9	New	117	60	104	99	41%	0	340
9	Cur.	146		133	128		47	4105
10	New	107	83	103	98	69%	0	306
10	Cur.	120		111	106		23	3891
11	New	127	76	118	113	46%	0	310
11	Cur.	164		142	137		50	4109
12	New	134	76	123	118	45%	0	408
12	Cur.	170		148	142		53	4105
13	New	131	81	124	119	47%	0	295
13	Cur.	173		144	139		49	4107
14	New	111	73	105	100	53%	0	247
14	Cur.	138		128	123		36	3223
15	New	96	57	100	95	46%	0	236
15	Cur.	126		117	112		32	3149

Instance 2, $\omega = 4$

Sim	Method	AC	AC r.	ND	ND30	Cost%	ENNR	HC
Ave	New	165.9	137.2	31.2	31.2	57%	0	385.5
Ave	Cur.	244.1		41.8	41.6		7.3	No Arr.
1	New	188.1	155.2	34	34	59%	0	355
1	Cur.	264.6		44	44		10	No Arr.
2	New	123.2	103.5	26	26	57%	0	300
2	Cur.	180.8		35	34		5	No Arr.
3	New	161.2	149.1	31	31	71%	0	300
3	Cur.	211.2		39	39		4	No Arr.
4	New	153.1	139.6	30	30	65%	0	366
4	Cur.	215.3		35	35		3	No Arr.
5	New	174.2	150.4	32	32	65%	0	348
5	Cur.	232.9		39	39		7	No Arr.
6	New	185.3	157.0	34	34	58%	0	436
6	Cur.	270.0		45	45		8	No Arr.
7	New	126.8	93.6	24	24	42%	0	303
7	Cur.	224.0		39	39		8	No Arr.
8	New	131.2	108.2	24	24	58%	0	300
8	Cur.	185.3		35	35		6	No Arr.
9	New	181.7	160.1	33	33	67%	0	333
9	Cur.	237.3		42	42		7	No Arr.
10	New	196.3	159.1	33	33	55%	0	540
10	Cur.	291.5		47	47		8	No Arr.
11	New	143.0	121.6	29	29	67%	0	300
11	Cur.	182.3		37	37		6	No Arr.
12	New	183.9	136.9	36	36	46%	0	540
12	Cur.	297.4		47	47		10	No Arr.
13	New	181.9	135.1	29	29	44%	0	436
13	Cur.	305.1		47	47		11	No Arr.
14	New	150.5	130.1	32	32	54%	0	300
14	Cur.	239.5		44	44		6	No Arr.
15	New	185.1	142.3	39	39	54%	0	540
15	Cur.	262.7		47	45		9	No Arr.

Instance 3, $\omega = 4$

Sim	Method	AC	AC r.	ND	Cost%	ENNR	HC
Ave	New	181.3	130.6	63.5	58.7%	0.07	444.6
Ave	Cur.	221.6		77.5		23.5	No Arr.
1	New	194.9	147.9	70	60%	0	540
1	Cur.	246.9		85		23.0	No Arr.
2	New	184.2	136.1	69	66%	0	540
2	Cur.	207.5		74		21.0	No Arr.
3	New	216.4	158.9	74	64%	0	579
3	Cur.	249.5		84		29.0	No Arr.
4	New	186.7	151.1	69	68%	0	579
4	Cur.	223.0		75		19.0	No Arr.
5	New	144.1	101.8	53	62%	0	325
5	Cur.	163.2		66		21.0	No Arr.
6	New	214.4	149.4	64	62%	0	540
6	Cur.	239.2		79		28.0	No Arr.
7	New	130.1	84.9	46	42%	0	300
7	Cur.	202.8		75		21.0	No Arr.
8	New	150.8	131.6	53	65%	0	348
8	Cur.	202.6		68		10.0	No Arr.
9	New	165.6	114.5	64	53%	0	388
9	Cur.	217.2		78		25.0	No Arr.
10	New	184.7	141.6	65	54%	0	360
10	Cur.	261.7		85		24.0	No Arr.
11	New	252.5	163.1	73	62.2%	1	No Arr.
11	Cur.	262.3		83		23.0	No Arr.
12	New	167.5	111.1	63	49%	0	436
12	Cur.	227.9		79		25.0	No Arr.
13	New	140.9	77.2	49	41%	0	366
13	Cur.	187.8		73		27.0	No Arr.
14	New	215.9	173.6	81	69%	0	381
14	Cur.	251.5		88		30.0	No Arr.
15	New	171.1	115.7	60	64%	0	543
15	Cur.	181.5		70		26.0	No Arr.

Instance 4, $\omega = 4$

Sim	Method	AC	AC r.	ND	ND30	Cost%	ENNR	HC
Ave	New	179.8	150.3	67.2	66.9	63%	0	475.6
Ave	Cur.	239.7		81.0	80.6		14.3	No Arr.
1	New	200.4	177.9	77	77	66%	0	436
1	Cur.	270.2		88	88		13	No Arr.
2	New	178.1	144.1	66	66	64%	0	540
2	Cur.	226.3		80	79		15	No Arr.
3	New	194.1	171.4	74	74	63%	0	540
3	Cur.	270.6		84	84		10	No Arr.
4	New	172.2	133.6	66	66	62%	0	540
4	Cur.	216.8		76	76		17	No Arr.
5	New	166.9	121.2	62	60	55%	0	540
5	Cur.	220.3		82	81		21	No Arr.
6	New	178.3	154.0	66	66	63%	0	471
6	Cur.	244.8		82	81		12	No Arr.
7	New	166.4	151.5	62	62	72%	0	355
7	Cur.	211.7		73	72		9	No Arr.
8	New	186.1	145.4	65	65	59%	0	540
8	Cur.	248.1		83	82		17	No Arr.
9	New	190.8	164.9	69	69	68%	0	540
9	Cur.	243.3		80	80		11	No Arr.
10	New	168.8	145.6	64	64	62%	0	436
10	Cur.	233.7		76	76		11	No Arr.
11	New	159.6	125.6	57	57	60%	0	436
11	Cur.	208.9		71	71		16	No Arr.
12	New	175.9	134.0	64	64	55%	0	436
12	Cur.	244.3		84	84		21	No Arr.
13	New	188.2	160.8	76	74	58%	0	377
13	Cur.	276.3		91	90		17	No Arr.
14	New	175.2	152.3	65	65	66%	0	436
14	Cur.	230.4		78	78		12	No Arr.
15	New	175.4	148.0	65	65	68%	0	471
15	Cur.	219.1		77	77		14	No Arr.

Instance 5, $\omega = 16$

Sim	Method	AC	AC r.	ND	Cost%	ENNR	HC	LC
Ave	New	273.8	122.4	68.0	31%	0	521.6	0.0
Ave	Cur.	398.7		91.4		42.4	No Arr.	102.9
1	New	278.3	102.5	66	24%	0	543	0
1	Cur.	421.5		94		45	No Arr.	0
2	New	191.0	89.4	54	31%	0	480	0
2	Cur.	291.5		71		26	No Arr.	0
3	New	295.4	151.0	74	37%	0	491	0
3	Cur.	408.5		95		45	No Arr.	0
4	New	288.9	153.8	69	37%	0	515	0
4	Cur.	419.5		95		41	No Arr.	0
5	New	330.4	205.8	84	46%	0	543	0
5	Cur.	444.2		95		44	No Arr.	0
6	New	272.4	65.1	67	14%	0	579	0
6	Cur.	459.0		100		53	No Arr.	480
7	New	281.3	100.1	81	22%	0	543	0
7	Cur.	450.0		100		53	No Arr.	480
8	New	249.2	117.8	61	33%	0	480	0
8	Cur.	361.8		89		37	No Arr.	0
9	New	283.8	164.3	69	44%	0	480	0
9	Cur.	375.9		88		38	No Arr.	0
10	New	286.3	114.0	64	27%	0	540	0
10	Cur.	417.0		93		45	No Arr.	0
11	New	317.0	172.1	82	38%	0	543	0
11	Cur.	450.7		100		48	No Arr.	480
12	New	255.7	80.9	66	21%	0	540	0
12	Cur.	383.8		89		43	No Arr.	0
13	New	265.3	116.6	69	32%	0	543	0
13	Cur.	368.7		88		40	No Arr.	0
14	New	290.7	108.6	71	24%	0	483	0
14	Cur.	450.1		100		50	No Arr.	480
15	New	264.2	144.1	57	44%	0	543	0
15	Cur.	330.0		82		34	No Arr.	0

Instance 6, $\omega = 16$

Sim	Method	AC	AC r.	ND	ND30	Cost%	ENNR	HC
Ave	New	273.3	93.8	136.5	135.9	25%	0.1	780.3
Ave	Cur.	367.3		176.4	176.4		94.3	No Arr.
1	New	301.1	91.6	128	128	27%	2	No Arr.
1	Cur.	335.2		167	167		86	No Arr.
2	New	306.3	143.0	163	163	40%	0	548
2	Cur.	357.8		175	175		97	No Arr.
3	New	337.1	176.9	163	163	43%	0	579
3	Cur.	413.7		188	188		103	No Arr.
4	New	235.9	51.8	119	119	15%	0	480
4	Cur.	355.4		176	176		95	No Arr.
5	New	204.7	72.1	112	112	27%	0	480
5	Cur.	266.7		143	143		65	No Arr.
6	New	267.2	77.0	130	130	20%	0	491
6	Cur.	384.5		181	181		96	No Arr.
7	New	280.3	60.0	153	143	14%	0	550
7	Cur.	416.4		189	189		108	No Arr.
8	New	268.4	89.3	136	136	25%	0	550
8	Cur.	361.1		176	176		91	No Arr.
9	New	305.1	98.5	140	140	24%	0	560
9	Cur.	411.2		188	188		106	No Arr.
10	New	269.2	109.1	140	140	30%	0	517
10	Cur.	366.9		180	180		93	No Arr.
11	New	273.8	92.9	145	145	25%	0	548
11	Cur.	368.6		178	178		98	No Arr.
12	New	210.6	51.9	94	94	17%	0	508
12	Cur.	300.1		155	155		74	No Arr.
13	New	282.6	88.5	147	147	22%	0	491
13	Cur.	405.2		187	187		107	No Arr.
14	New	259.7	82.2	117	117	23%	0	543
14	Cur.	352.8		173	173		90	No Arr.
15	New	297.5	122.3	161	161	30%	0	540
15	Cur.	413.5		190	190		105	No Arr.

Instance 7, $\omega = 40$

Sim	Method	AC	AC r.	ND	ND30	Cost%	ENNR	HC
Ave	New	241.1	117.2	134.5	134.5	38%	0	478.7
Ave	Cur.	312.0		174.5	174.5		83.9	No Arr.
1	New	249.2	108.8	136	136	30%	0	515
1	Cur.	360.6		186	186		93	No Arr.
2	New	255.9	140.4	155	155	40%	0	400
2	Cur.	350.7		189	189		93	No Arr.
3	New	208.9	77.3	128	128	27%	0	429
3	Cur.	289.2		170	170		80	No Arr.
4	New	263.9	177.4	144	144	67%	0	543
4	Cur.	265.6		153	152		65	No Arr.
5	New	225.6	95.2	121	121	31%	0	400
5	Cur.	310.0		177	177		86	No Arr.
6	New	250.4	138.1	139	139	45%	0	437
6	Cur.	305.4		176	176		85	No Arr.
7	New	258.1	159.2	151	151	58%	0	543
7	Cur.	274.0		163	163		75	No Arr.
8	New	282.3	190.5	158	158	57%	0	436
8	Cur.	336.1		180	180		82	No Arr.
9	New	226.9	68.9	115	115	22%	0	579
9	Cur.	314.3		175	175		89	No Arr.
10	New	218.7	141.1	124	124	58%	0	491
10	Cur.	243.6		147	147		58	No Arr.
11	New	257.7	117.4	141	141	34%	0	501
11	Cur.	348.5		188	188		98	No Arr.
12	New	251.2	108.4	133	133	33%	0	579
12	Cur.	326.9		177	177		89	No Arr.
13	New	248.7	93.1	133	133	26%	0	501
13	Cur.	363.9		189	189		96	No Arr.
14	New	223.8	101.3	116	116	35%	0	427
14	Cur.	290.6		169	169		81	No Arr.
15	New	194.6	41.2	123	123	14%	0	400
15	Cur.	300.0		179	179		89	No Arr.

Passengers moved off delayed flight in first stage

Sim	Ins 2	Ins 3	Ins 4	Ins 5	Ins 6	Ins 7
Ave	12.7	17.1	17.4	37.1	69.9	52.1
1	12	15	15	43	62	66
2	13	15	18	38	71	54
3	8	15	16	36	53	54
4	13	11	17	35	80	33
5	10	13	20	36	67	56
6	11	15	20	48	70	49
7	18	29	13	49	80	24
8	11	15	21	31	72	32
9	11	23	15	28	74	61
10	14	21	16	37	61	60
11	8	21	18	30	80	48
12	18	27	22	44	65	57
13	18	15	22	34	83	68
14	12	12	15	42	60	53
15	14	10	13	26	71	67

APPENDIX C

Run Time Data

Instance 1 run time (seconds)

# Fts	MIP	Bend	Bend, p	B/MIP	B,p/MIP	# Its
6	0.06	0.44	0.34	703%	553%	2

Instance 2 run time (seconds)

Sim	# Fts	MIP	Bend	Bend, p	B/MIP	B,p/MIP	# Its
Ave	96.3	7.6	10.6	6.3	141%	83%	2.0
1	103	7.2	10.3	6.2	143%	86%	2
2	88	6.4	8.2	4.6	128%	73%	2
3	105	6.4	10.0	5.9	156%	92%	2
4	85	8.7	12.2	6.7	140%	77%	2
5	84	7.2	10.2	6.0	142%	83%	2
6	80	7.1	10.3	6.2	146%	87%	2
7	117	7.2	10.7	6.3	149%	87%	2
8	87	6.7	10.7	6.4	160%	96%	2
9	100	8.6	10.1	5.6	117%	65%	2
10	112	9.3	10.3	6.5	111%	70%	2
11	92	7.5	11.8	7.2	158%	96%	2
12	90	7.8	11.3	6.8	145%	88%	2
13	88	6.9	9.7	5.7	141%	83%	2
14	104	8.2	11.9	7.1	146%	87%	2
15	109	8.3	11.3	6.7	135%	80%	2

Instance 3 run time (seconds)

Sim	# Fts	MIP	Bend	Bend, p	B/MIP	B,p/MIP	# Its
Ave	100.5	7.0	19.6	10.3	282%	147%	4.9
1	68	6.7	28.2	14.2	423%	214%	7
2	85	6.8	19.8	10.2	292%	151%	5
3	119	7.8	19.4	10.2	248%	130%	5
4	109	7.2	26.2	13.8	366%	193%	6
5	78	6.5	11.7	6.1	179%	93%	3
6	132	6.9	11.9	6.3	173%	91%	3
7	75	6.6	22.9	11.9	349%	181%	6
8	105	7.0	12.0	6.4	170%	91%	3
9	73	6.7	19.1	9.9	285%	148%	5
10	123	7.2	16.3	8.7	226%	121%	4
11	112	6.8	19.2	9.9	281%	145%	5
12	123	6.9	23.0	11.9	334%	173%	6
13	107	7.4	16.1	8.5	219%	116%	4
14	105	7.5	21.4	11.8	286%	157%	5
15	93	6.8	26.7	14.1	393%	207%	6

Instance 4 run time (seconds)

Sim	# Fts	MIP	Bend	Bend, p	B/MIP	B,p/MIP	# Its
Ave	155.2	7.9	11.9	7.7	150%	97%	2.1
1	164	8.1	16.0	12.2	198%	151%	2
2	148	7.3	10.2	6.3	139%	86%	2
3	148	7.3	11.9	7.4	164%	102%	2
4	116	10.8	13.5	9.0	125%	84%	2
5	171	7.9	10.8	6.9	138%	88%	2
6	174	8.0	11.7	7.5	146%	94%	2
7	160	7.1	10.4	6.6	146%	92%	2
8	164	7.5	10.7	6.7	143%	90%	2
9	172	7.3	10.5	6.6	144%	90%	2
10	132	8.1	12.2	7.8	151%	97%	2
11	113	7.8	15.2	9.3	195%	119%	3
12	174	8.2	11.3	7.0	137%	85%	2
13	167	7.9	11.1	7.0	141%	90%	2
14	180	8.0	11.5	7.5	143%	94%	2
15	145	7.8	10.8	6.9	138%	88%	2

Instance 5 run time (seconds)

Sim	# Fts	MIP	Bend	Bend, p	B/MIP	B,p/MIP	# Its
Ave	94.9	28.0	36.6	14.4	131%	52%	2.1
1	94	27.0	36.9	15.7	137%	58%	2
2	106	31.9	33.9	12.9	106%	41%	2
3	104	29.7	36.0	14.2	121%	48%	2
4	92	27.4	35.3	14.3	129%	52%	2
5	68	25.8	32.4	11.7	126%	46%	2
6	106	28.2	31.9	12.4	113%	44%	2
7	82	26.1	31.8	12.6	122%	48%	2
8	90	26.7	33.2	13.1	124%	49%	2
9	95	28.3	65.5	25.1	231%	89%	4
10	95	29.6	35.6	14.2	120%	48%	2
11	103	30.8	34.1	13.0	111%	42%	2
12	91	27.9	35.6	13.8	128%	50%	2
13	104	25.8	34.8	14.7	135%	57%	2
14	75	28.3	33.9	13.2	120%	47%	2
15	119	27.3	38.0	15.1	139%	55%	2

Instance 6 run time (seconds)

Sim	# Fts	MIP	Bend	Bend, p	# Its
Ave	92.2	n/a	138.3	62.7	6.6
1	75	n/a	124.0	55.1	6
2	91	n/a	94.0	39.3	5
3	99	n/a	n/a	n/a	n/a
4	84	n/a	75.0	31.1	4
5	99	n/a	117.3	51.8	6
6	104	n/a	194.5	93.8	9
7	83	n/a	139.4	60.4	7
8	83	n/a	101.6	44.1	5
9	98	n/a	150.1	67.5	7
10	93	n/a	172.7	80.7	8
11	113	n/a	180.2	84.2	8
12	85	n/a	n/a	n/a	n/a
13	100	n/a	99.5	42.5	5
14	92	n/a	247.4	120.9	11
15	84	n/a	102.2	44.3	5

Instance 7 run time (seconds)

Sim	# Fts	MIP	Bend	Bend, p	# Its
Ave	94.2	n/a	186.7	87.1	3.2
1	92	n/a	483.7	256.0	7
2	108	n/a	122.9	50.7	2
3	98	n/a	100.7	42.0	2
4	101	n/a	100.2	42.8	2
5	99	n/a	158.7	71.5	3
6	83	n/a	193.1	86.7	3
7	79	n/a	286.1	136.1	5
8	114	n/a	117.1	52.6	2
9	76	n/a	224.0	102.8	4
10	80	n/a	167.9	74.0	3
11	93	n/a	383.1	192.3	6
12	105	n/a	159.1	69.9	3
13	96	n/a	104.0	45.1	2
14	100	n/a	100.5	42.3	2
15	89	n/a	99.6	41.2	2

BIBLIOGRAPHY

BIBLIOGRAPHY

Abdelghany, A., G. Ekollu, R. Narasimhan, and K. Abdelghany (2004), A proactive crew recovery decision support tool for commercial airlines during irregular operations, *Annals of Operations Research*, *127*(1), 309–331.

AhmadBeygi, S., A. Cohn, Y. Guan, and P. Belobaba (2008), Analysis of the potential for delay propagation in passenger airline networks, *Journal of Air Transport Management*, *14*(5), 221–236.

AhmadBeygi, S., A. Cohn, and M. Lapp (2010), Decreasing airline delay propagation by re-allocating scheduled slack, *IIE Transactions*, *42*(7), 478–489.

Appelgren, L. (1969), A column generation algorithm for a ship scheduling problem, *Transportation Science*, *3*(1), 53–68.

Applegate, D., L. Breslau, and E. Cohen (2004), Coping with network failures: Routing strategies for optimal demand oblivious restoration, in *ACM SIGMETRICS performance evaluation review*, vol. 32, pp. 270–281, ACM.

Barnhart, C., and A. Cohn (2004), Commissioned paper airline schedule planning: Accomplishments and opportunities, *Manufacturing & service operations management*, *6*(1), 3–22.

Barnhart, C., E. Johnson, G. Nemhauser, M. Savelsbergh, and P. Vance (1998), Branch-and-price: Column generation for solving huge integer programs, *Operations Research*, pp. 316–329.

Barnhart, C., T. Kniker, and M. Lohatepanont (2002), Itinerary-based airline fleet assignment, *Transportation Science*, *36*(2), 199.

Barnhart, C., A. Cohn, E. Johnson, D. Klabjan, G. Nemhauser, and P. Vance (2003), Airline crew scheduling, *Handbook of Transportation Science*, pp. 517–560.

Belobaba, P. (1989), Application of a probabilistic decision model to airline seat inventory control, *Operations Research*, pp. 183–197.

Belobaba, P., A. Odoni, and C. Barnhart (2009), *The global airline industry*, Wiley.

Bertsimas, D., and J. N. Tsitsiklis (1997), *Introduction to Linear Optimization*, Athena Scientific, Post Office Box 805, Nashua, NH 03061, U.S.A.

Bratu, S., and C. Barnhart (2006), Flight operations recovery: New approaches considering passenger recovery, *Journal of Scheduling*, *9*(3), 279–298.

Burke, E., P. De Causmaecker, G. De Maere, J. Mulder, M. Paelinck, and G. Vanden Berghe (2010), A multi-objective approach for robust airline scheduling, *Computers & Operations Research*, *37*(5), 822–832.

Cao, J., and A. Kanafani (1997a), Real-time decision support for integration of airline flight cancellations and delays part i: mathematical formulation, *Transportation Planning and Technology*, *20*(3), 183–199.

Cao, J., and A. Kanafani (1997b), Real-time decision support for integration of airline flight cancellations and delays part ii: algorithm and computational experiments, *Transportation Planning and Technology*, *20*(3), 201–217.

Carlson, P. (2000), Exploiting the opportunities of collaborative decision making: A model and efficient solution algorithm for airline use, *Transportation Science*, *34*(4), 381–393.

Clarke, M., and B. Smith (2004), Impact of operations research on the evolution of the airline industry, *Journal of aircraft*, *41*(1), 62–72.

Clarke, M., et al. (1998), Development of heuristic procedures for flight rescheduling in the aftermath of irregular airline operations, *Tech. rep.*, [Cambridge, Mass.: Massachusetts Institute of Technology], Flight Transportation Laboratory,[1998].

Clausen, J., J. Hansen, A. Larsen, and J. Larsen (2001), Disruption management: operations research between planning and execution.

Cohn, A. (2009), Ioe 591, university of Michigan, Ann Arbor, Michigan.

Cohn, A., and C. Barnhart (2003), Improving crew scheduling by incorporating key maintenance routing decisions, *Operations Research*, pp. 387–396.

Cordeau, J., G. Stojkovic, F. Soumis, and J. Desrosiers (2001), Benders decomposition for simultaneous aircraft routing and crew scheduling, *Transportation Science*, *35*(4), 375.

Eggenberg, N., M. Salani, and M. Bierlaire (2010), Constraint-specific recovery network for solving airline recovery problems, *Computers & Operations Research*, *37*(6), 1014–1026.

Ehrgott, M., and D. Ryan (2002), Constructing robust crew schedules with bicriteria optimization, *Journal of Multi-Criteria Decision Analysis*, *11*(3), 139–150.

Fuhr, B. (2007), Robust flight scheduling-an analytic approach to performance evaluation and optimization.

Garrow, L. (2009), Online travel data: A goldmine of new opportunities, *Journal of Revenue & Pricing Management*, *8*(2), 247–254.

Hane, C., C. Barnhart, E. Johnson, R. Marsten, G. Nemhauser, and G. Sigismondi (1995), The fleet assignment problem: solving a large-scale integer program, *Mathematical Programming*, *70*(1), 211–232.

Hsiao, C., and M. Hansen (2006), Econometric analysis of us airline flight delays with time-of-day effects, *Transportation Research Record: Journal of the Transportation Research Board*, *1951*(-1), 104–112.

http://www.transtats.bts.gov (), Bureau of transportation statistics.

Jacobs, T., B. Smith, and E. Johnson (2008), Incorporating network flow effects into the airline fleet assignment process, *Transportation Science*, *42*(4), 514–529.

Jafari, N., and S. Hessameddin Zegordi (2010), Simultaneous recovery model for aircraft and passengers, *Journal of the Franklin Institute*.

Janić, M. (2005), Modeling the large scale disruptions of an airline network, *Journal of transportation engineering*, *131*, 249.

Jarrah, A., G. Yu, N. Krishnamurthy, and A. Rakshit (1993), A decision support framework for airline flight cancellations and delays, *Transportation Science*, *27*(3), 266–280.

Johnson, E. (1989), Modelling and strong linear programs for mixed integer programming, in *Algorithms and Model Formulations in Mathematical Programming*, pp. 1–43, Springer-Verlag New York, Inc.

Jonathan, F., Y. Gang, and F. Michael (2001), Optimizing aircraft routings in response to groundings and delays, *IIE Transactions*, *33*(10), 931–947.

Kohl, N., and S. Karisch (2004), Airline crew rostering: Problem types, modeling, and optimization, *Annals of Operations Research*, *127*(1), 223–257.

Kohl, N., A. Larsen, J. Larsen, A. Ross, and S. Tiourine (2007), Airline disruption management–Perspectives, experiences and outlook, *Journal of Air Transport Management*, *13*(3), 149–162.

Lan, S., J. Clarke, and C. Barnhart (2006), Planning for robust airline operations: Optimizing aircraft routings and flight departure times to minimize passenger disruptions, *Transportation Science*, *40*(1), 15–28.

Lapp, M., S. AhmadBeygi, A. Cohn, and O. Tsimhoni (2008), A recursion-based approach to simulating airline schedule robustness, in *Proceedings of the 40th Conference on Winter Simulation*, pp. 2661–2667, Winter Simulation Conference.

Lettovsky, L. (1997), Airline operations recovery: An optimization approach.

Lettovsky, L., E. Johnson, and G. Nemhauser (2000), Airline crew recovery, *Transportation Science*, *34*(4), 337.

Lohatepanont, M., and C. Barnhart (2004), Airline schedule planning: Integrated models and algorithms for schedule design and fleet assignment, *Transportation Science*, *38*(1), 19–32.

Luo, S., and G. Yu (1997), On the airline schedule perturbation problem caused by the ground delay program, *Transportation Science*, *31*, 298–311.

Mercier, A., and F. Soumis (2007), An integrated aircraft routing, crew scheduling and flight retiming model, *Computers & Operations Research*, *34*(8), 2251–2265.

Mercier, A., J. Cordeau, and F. Soumis (2005), A computational study of benders decomposition for the integrated aircraft routing and crew scheduling problem, *Computers & Operations Research*, *32*(6), 1451–1476.

Petersen, J., G. Solveling, E. Johnson, J. Clarke, and S. Shebalov (2010), An Optimization Approach to Airline Integrated Recovery.

Rexing, B. (1997), Airline fleet assignment with time windows, Ph.D. thesis, Massachusetts Institute of Technology.

Rexing, B., C. Barnhart, T. Kniker, A. Jarrah, and N. Krishnamurthy (2000), Airline fleet assignment with time windows, *Transportation Science*, *34*(1), 1–20.

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

Rosenberger, J., E. Johnson, and G. Nemhauser (2003), Rerouting aircraft for airline recovery, *Transportation Science*, *37*(4), 408.

Ryan, D., and B. Foster (1981), An integer programming approach to scheduling, *Computer scheduling of public transport urban passenger vehicle and crew scheduling*, pp. 269–280.

Schaefer, A., E. Johnson, A. Kleywegt, and G. Nemhauser (2005), Airline crew scheduling under uncertainty, *Transportation Science*, *39*(3), 340–348.

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

Van Roy, B., and K. Mason (2005), Formulation and Analysis of Linear Programs.

Yan, S., and D. Yang (1996), A decision support framework for handling schedule perturbation, *Transportation Research Part B: Methodological*, *30*(6), 405–419.

Yen, J., and J. Birge (2006), A stochastic programming approach to the airline crew scheduling problem, *Transportation Science*, *40*(1), 3–14.

Yu, G., and X. Qi (2004), *Disruption management: framework, models and applications*, World Scientific Pub Co Inc.