Degradable Airline Scheduling: an Approach to Improve Operational Robustness and Differentiate Service Quality

by

Laura Sumi Kang

B.E., Korea University (1999)

Submitted to the Sloan School of Management
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Operations Research

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2004

Author ....
Sloan School of Management
December 18, 2003

Certified by ....
John-Paul B. Clarke
Associate Professor, Department of Aeronautics and Astronautics
Thesis Supervisor

Accepted by ....
James B. Orlin
Edward Pennell Brooks Professor of Operations Research
Co-director, Operations Research Center

Degradable Airline Scheduling: an Approach to Improve Operational Robustness and Differentiate Service Quality

by

Laura Sumi Kang

Submitted to the Sloan School of Management
on December 18, 2003, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Operations Research

Abstract

We present a methodology for deriving robust airline schedules that are not vulnerable to disruptions caused by bad weather. In this methodology, the existing schedule is partitioned into independent sub-schedules or layers – prioritized on the basis of revenue – that provide airlines with a clear delay/cancellation policy and may enable them to market and sell tickets for flight legs based on passenger preference for reliability.

We present three different ways to incorporate degradability into the scheduling process: (1) between flight scheduling and fleet assignment (degradable schedule partitioning model), (2) with fleet assignment (degradable fleet assignment model), and (3) with aircraft routing (degradable aircraft routing model). Each problem is modeled as an integer program. Search algorithms are applied to the degradable aircraft routing model, which has a large number of decision variables.

Results indicate that we can successfully assign flight legs with high revenue itineraries in the higher priority layer without adding aircraft or changing the schedule, and differentiate the service quality for passengers in different priority layers. Passengers in the high priority layers have much less delay and fewer cancellations than passengers in low priority layers even during the bad weather. In terms of recovery cost, which includes revenue lost, operational cost saving and crew delay cost, degradable airline schedules can save up to $30,000 per day. Degradable airline schedules have cost saving effect, especially when an airport with a high capacity reduction in bad weather is affected by bad weather.

Thesis Supervisor: John-Paul B. Clarke
Title: Associate Professor, Department of Aeronautics and Astronautics

Acknowledgments

I cannot possibly express my appreciation for my thesis advisor, Professor John-Paul Clarke. He has been a great advisor, teacher, and friend. He gave me lots of priceless opportunities to learn and grow. With his support and encouragement, now I am a better researcher and a stronger person. I would like to acknowledge the rest of my committee, Professor Cynthia Barnhart, Professor James Orlin, and Dr. Michael Clarke. I was extremely lucky to have such a diverse committee. Professor Barnhart helped me to build models and gave valuable feedback throughout the whole process. Professor Orlin's advice on search algorithms was helpful in developing and implementing them. Dr. Clarke gave me insights about the real-world operations and constraints.

Terran Melconian, the creator of MEANS, helped me implement our solutions to MEANS. Chapter 6 would not exist without him. I also would like to thank Professor Georgia Perakis and Professor Andreas Schulz for their moral support throughout my years at MIT.

Without question, my family and friends have been my biggest source of support. I especially thank my parents for always trusting me, supporting me, and praying for me even though we are on opposite sides of the world. I cannot thank them enough for having encouraged me to come to the U.S. for graduate school. I am thankful for so many tremendous friends in my life. My old friends, who probably still do not understand exactly what I am doing, have been a great hot-line for my many ups and downs. I especially thank my friends at ORC and ICAT for keeping my everyday life enjoyable. They are what I will miss the most.

Last but not least, I would like to extend my thanks to Professor Youngho Lee, who introduced me to Operations Research, for his continous and endless support and advice. Without him, I wouldn't be here.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

When any part of a system fails to work properly, it often affects the performance of the entire system. One approach to increase system robustness is to design systems with independent sub-systems that may be operated independently. By partitioning the system into independent sub-systems, we are able to isolate the impact of poor performance in any one sub-system, and restrict any errors to the sub-system where it originally occurred. Independent sub-systems allow us to differentiate the level of performance for each sub-system simply by prioritizing them.

A degradable system is defined as a system which is composed of several independent sub-systems, and each sub-system has been assigned a different priority. The idea of a degradable system can have various applications, such as those in telecommunication, scheduling, or transportation. A performance criterion for prioritizing depends on the application area and needs of users. In this thesis, we apply this concept to the airline industry in order to explore the implementation possibility and potential advantages of a degradable system with different levels of reliability.

The structure of the paper is as follows: In the rest of this chapter, we state the problem and propose a solution. We introduce the concept of degradable airline scheduling, and define different cases that we will discuss for the rest of the thesis. In Chapter 2, we present the integer programming models for each case defined. In Chapter 3, we provide the details of a prototype application of our models to a representative U.S. domestic network airline. Solution approaches used for each case are presented in Chapter 4, followed by a discussion of the model results and simulation results in Chapters 5 and 6, respectively. The sensitivity analysis is presented in Chapter 7. We then review our results and discuss further research in Chapter 8.

1.1 Background

The capacity of an airport is, in large part, a function of the prevailing weather conditions at that airport. When the visibility around an airport is good, the pilot of an aircraft may request a visual clearance, and if approved, aircraft operations may then be conducted according to visual flight rules (VFR). Under VFR, which have no formal separation requirements, the pilot is responsible for maintaining an appropriate separation from the preceding aircraft. When the visibility around an airport is poor, however, all aircraft operations must be conducted according to instrument flight rules (IFR). Under IFR, the air traffic controller is responsible for ensuring that the separation between the aircraft under his/her control is never less than pre-determined conservative minima. Because the separation required under IFR is greater than that typically observed under VFR, the capacity under IFR is therefore less than the capacity under VFR.

Airport capacity is also a function of wind speed and direction. For example, at an airport with independent parallel runways in one direction and a single runway in the perpendicular direction, a shift in wind direction of 90 degrees can cause a 50% reduction in arrival capacity and a similar reduction in departure capacity.

Whatever the cause, when airport capacity is reduced, airlines must delay or cancel flight legs to ensure that aircraft and crew are in position for later flights.

The impact of delays and cancellations on airlines and passengers can be significant. When flight legs are delayed, airlines incur the cost of the extra fuel that is burned during airborne delays or while aircraft wait on the airport surface with their engines running. In addition, airlines must pay crews for the extra time that they spend waiting on aircraft during delays in the air or on the ground. Because fuel and labor costs are two of the major costs incurred by an airline, delays can have significant financial ramifications. For example, it is estimated that the major US carriers currently lose as much as two billion dollars per year because of delays and cancellations [52].

Delays and cancellations can also have a significant negative impact on passengers. From a passenger's perspective, the most critical consideration is the difference between the scheduled and the actual arrival time at his/her ultimate destination. This 'final arrival delay' occurs because the last (or only) flight leg in an itinerary is delayed or cancelled, or because a connection between flight legs has been missed due to the delay or cancellation of one or more prior flight legs in their itinerary. Whatever the cause, this late arrival can result in considerable financial and personal loss to a passenger, and can also result in loss of goodwill toward the airline in question. Thus, both from a passenger and an airline perspective, it is very important to make airline operations as robust as possible to disruptions.

The traditional approach to dealing with such disruptions is to re-schedule aircraft and crew each time a disruption occurs. Because airlines must simultaneously consider changes to aircraft routings, crew schedules, and passenger itineraries, this approach is computationally challenging. Thus, airline recovery decision are most often made by airline operations controllers using heuristics that are based on years of experience. Even with these computational challenges, many researchers have developed airline recovery algorithms with the objective of the optimization models underlying these algorithms being to minimize the changes in schedule and/or minimize the delay and cancellation cost. Examples of these models are described below.

Yan and Tu [55] used a multi-commodity network flow model to reschedule multi-fleet aircraft routing to minimize the delay and cancellation cost. Bard et al. [6] used a minimum cost network flow problem to find aircraft routings that would minimize delay and cancellation cost. Thengvall et al. [53][54] built a multi-commodity network model for schedule recovery following a hub closure. This model allows not only cancellations and delays but also ferry flights and substitutions between fleets. They solved the problem with various objective functions, including minimizing cancellation and delay cost, and maintaining as much of the original routing as possible. Lettovsky [43] extracted a subset of the schedule for rescheduling to find a new crew schedule that would disturb the current schedule as little as possible. Heuristics to solve optimization models and extract a workable subset of the schedule are developed in order to get a recovery solution in real time. On the other hand, Bratu [15] focused more on passenger part. He proposed on-line models and algorithms to reduce passenger disruption and delays by cancelling and delaying flight departure.

However, even with the recent advances in the capabilities of recovery models, there is still one aspect of the airline recovery problem that will be difficult to overcome. At the time when the recovery decision is being made, airlines do not know the amount of delay that each passenger is willing to accept. Therefore, the existing solution can not be certain to minimize passenger displeasure. Even if the preferences of passengers were known, the mathematical model required to allocate delay as part of the recovery problem would be intractable, as the number of passengers far exceeds the number of aircraft and the number of crew. Thus, passengers will continue to feel that airlines are not being responsive to their needs.

A more proactive approach is to create airline schedules that intrinsically isolate or dampen the impact of disruptions so that any single delay or cancellation does not impact broadly the operations of the airline network. This may be achieved by creating schedules that are either easily recoverable from or less susceptible to schedule disruptions caused by bad weather or equipment failures. Therefore the fundamental step in this approach is to identify how the characteristics of a schedule affect the robustness of that schedule.

Ageeva and Clarke [3] introduced the idea that the flexibility and, by extension, the robustness of an airline schedule can be improved by increasing the number of overlapping routes in the solution to the aircraft maintenance routing problem so that aircraft can be easily swapped when disruptions occur. They recognized that much of the difficulty in deciding whether to switch an aircraft from its pre-assigned routing to another routing in order to cover a flight leg was due to the requirement that, in the resulting solution, the routings for all aircraft must include a maintenance opportunity before their next scheduled time for maintenance. They then postulated that the difficulty of the problem could be reduced if each aircraft routing "overlapped" at least twice with another aircraft routing between each maintenance visit. Chebalov and Klabjan [17][41] suggested a similar approach to the problem of crew scheduling where the robustness of a crew schedule would be improved if the opportunities for the switching of crews were increased. Rosenberger et al. [47] presented an optimization model that reduces the true operating cost of a schedule by incorporating the likely costs of rerouting or canceling a flight into the fleet assignment problem. Later they suggested a cancellation policy based on routes instead of individual flight legs. Building many short cycles in the schedule provides more robust aircraft rotations, which improve operation [48].

1.2 Degradable Airline Schedule

In this thesis, we introduce and develop the concept of degradable airline scheduling. A degradable schedule is one that is divided into several independent sub-schedules, or layers, and each layer has a different level of importance. The fundamental premise behind degradable airline schedules is that it is possible to simultaneously increase the robustness of an airline's operation and segment the product that is offered to passengers based on reliability, and thereby change the way airline tickets are sold. Specifically, the degradable airline schedule (1) is robust in response to disruptions due to weather at or around airports, (2) has itineraries with different reliability levels, (3) provides passengers with a mechanism to select flights based on their own preference for reliability, (4) provides airlines with a mechanism to increase customer satisfaction, and (5) provides the flexibility required to more easily respond to changes imposed on the airline by air traffic control.

A schedule that is partitioned into independent sub-schedules or layers is robust under disruptions for two reasons. First, because the aircraft routings within each layer are independent of the aircraft routings in other layers, a delay or cancellation within a layer can only impact flight legs within that layer and will not propagate to other layers. Thus, the impact of a delay or cancellation is limited, even if nothing is done to adjust the schedule. Second, because fewer flight legs, aircraft routings, and crew schedules are affected by a given delay or cancellation, re-optimization of the airline's operations is simpler and thus more effective.

The partitioning of the schedule also serves as a mechanism to assign different reliabilities to itineraries. The specifics of this mechanism are as follows: If layers are prioritized, and the order in which delays and cancellations are assigned to layers is the reverse of the order of priority, the priority of a layer becomes a proxy for the reliability of the itineraries and flight legs within that layer. If, further, the highest priority layer is designed to be operable in bad weather, an airline can "protect" the itineraries in the highest priority layer from capacity reductions due to bad weather.

The concept of differentiating service quality based on customer preference for reliability has been discussed for some time regarding the problem of electricity distribution. For example, Fumagalli et. al [26] suggested a reliability insurance scheme for the distribution of electricity to customers who require different levels of reliability. Specifically, they showed that consumers valued, and were willing to pay for, reliability insurance. They also showed that the addition of reliability insurance improved the overall customer satisfaction and increased the profit margin of the service provider. These findings are consistent with the often used axiom in customer service that overall satisfaction with a service provider is improved if consumers are provided with a product, or level of service, that is well matched to what they need, or perhaps more importantly, what they are willing to pay for.

This property of the degradable schedule also provides a mechanism for passengers to incorporate into their decision their delay threshold and their ideas about which itinerary to purchase. If passengers know what the levels of reliability of the itineraries are, they will have the information they need to weigh this information against cost, departure time, arrival time, travel time, ticket restrictions, and all the other factors that passengers currently consider. Thus, if ticket prices are also a function of the layer of the corresponding itineraries, passengers will be able to signal their preference through the price they are willing to pay. If the schedule is constructed in such a way that passengers know the reliability of the itineraries they are considering before they purchase their ticket, and the ticket prices reflect the reliability of the itineraries in question, then passengers will have a clear expectation of service quality and an economic mechanism for choosing their preference levels of reliability. This ability to simultaneously improve overall customer satisfaction and increase the profit margin of the service provider could be very beneficial to an industry that is noted for small profit margins in the best of times, and large losses in the worst of times. The challenge, then, is to develop a schedule in which the reliability of different itineraries can be quantified and guaranteed.

As previous research has shown, customer satisfaction is primarily a function of the difference between *the expected quality* of service and *the perceived quality* of service, not the quality of service itself [25]. Thus, by providing passengers with an accurate estimate of the delay they can expect when the weather deteriorates, the difference between the actual delay and the expected delay will be smaller, and their dissatisfaction will be reduced. This may also lead to increased revenues since it can be argued that customers will be willing to pay more for a service that has distinct benefits to them relative to existing operations.

In addition to the potentially enhanced customer satisfaction and increased revenue, the degradable airline schedule also provides flexibility required to more easily respond to changes in the air traffic control environment. Specifically, the clear prioritization of flight legs within the schedule is consistent with many of the market-based schemes that have been proposed for the allocation of capacity in the U.S. national airspace system. In fact, the priority for all flights in each airline's schedule can be used as the basis for the airport capacity auctions that have been proposed by both Milner [43] [45] and Hall [35] as mechanisms to efficiently allocate limited capacity.

1.3 Incorporating Degradability into the Scheduling Process

Because of its complexity and size, the airline scheduling problem is often decomposed into several smaller problems, solved in four sequential steps [34]. As Figure 1-1 shows, the four steps in the airline scheduling process are: flight scheduling, fleet assignment, aircraft routing, and crew scheduling.

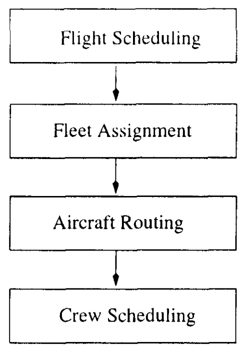

Figure 1-1: Airline Scheduling Process

Flight scheduling is usually done well in advance of the time that a flight occurs, and is based on marketing considerations, the planned flights of competitors, and internal airline considerations such as operating costs and network structure. The resulting flight schedule is a set of flights; each is composed of one or more flight legs. Each flight is characterized by a flight number, origin, destination, scheduled departure time, and scheduled arrival time. Given the flight schedule, a specific fleet type is assigned to each flight leg using *fleet assignment models*. Since each fleet type has a certain capacity and operating cost, an airline must decide on the fleet type for each flight leg based on predicted passenger demand, its given capacity, its operating cost, and the number of aircraft available. Once the fleet type for each flight leg is decided, *aircraft routing models* are used to assign specific aircraft to flight legs based on maintenance constraints while ensuring that each flight leg is flown by exactly one aircraft. Aircraft routing is solved for each fleet type, and different maintenance stations and maintenance requirements are applied for each fleet type. Finally, in *crew scheduling*, specific cockpit and cabin crews are assigned to each flight leg based on the crew's qualifications to operate different fleet types, and on workplace regulations such as how many hours a person may pilot an aircraft and how many people must make up the crew for a particular fleet.

Much of the recent work in airline scheduling has focused on improving schedule optimality by combining two or more steps in the airline scheduling process. Rexing et. al [46] introduced the idea of adding time windows to the fleet assignment problem so that departure times can be adjusted if small adjustments will increase the optimality of the solution. This approach incorporates a component of the flight scheduling problem (i.e. setting the departure time of flights) into the fleet assignment problem. Clarke et. al [19] introduced maintenance and crew consideration into fleet management. Barnhart et al. [7] solved fleet assignment with aircraft routing. Cohn and Barnhart [21] combined crew scheduling with aircraft routing, including maintenance considerations. Cordeau et al. [22] also solved aircraft routing and crew scheduling simultaneously. Lohatepanont and Barnhart [44] integrated schedule design and fleet assignment.

As Figure 1-2 shows, the concept of degradable scheduling could be introduced into the airline scheduling process (a) between the flight scheduling and fleet assignment, (b) in combination with fleet assignment, and (c) in combination with aircraft routing.

Degradable Schedule Partition Model (D-SPM) is used to assign flight legs to different layers. Given the flight schedule, D-SPM simply partitions the schedule into several sets where each set of scheduling satisfies balance constraints. Fleet assignment, aircraft routing, and crew scheduling are completed for each layer. Degradable Fleet Assignment Model (D-FAM) is used to solve degradable airline scheduling and fleet assignment models simultaneously. Instead of only deciding which layer the flight leg should be in, D-FAM also decides which fleet type should be used as well. Aircraft routing and crew scheduling are completed separately for each fleet-layer

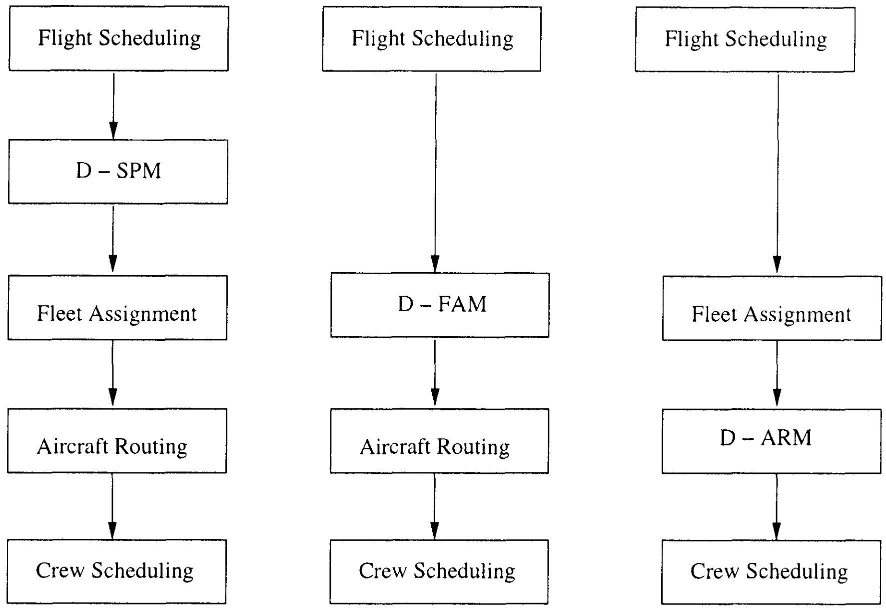

Figure 1-2: Potential Locations for Degradable Airline Scheduling

network. Degradable Aircraft Routing Model (D-ARM) is designed to solve degradable airline scheduling and aircraft routing simultaneously given the chosen flight schedule and fleet assignment. D-ARM builds a route for each aircraft by connecting the sequence of flight legs with the same fleet type, and then assigns the route to a proper layer. There are pros and cons of each model. We can add more robustness, or degradability, to the model as we consider degradable scheduling earlier in the planning process. However, this may restrict the fleet assignment and/or aircraft routing, thus result in higher operating cost. In this thesis, with a given schedule, we only solve fleet assignment and aircraft routing, but not crew scheduling.

Chapter 2

Modeling Approaches

In the ideal degradable schedule, itineraries would be assigned to layers in order of importance; that is, the set of itineraries in any layer would be the same as the set of itineraries derived by ranking itineraries in order of importance and then partitioning them into layers. This would be easy to achieve for a purely point-to-point airline that has non-stop flights between every origin and destination, and only sells tickets for single flight leg itineraries. In this case, each itinerary would have a unique flight leg, and assigning flight legs to layers in order of importance would give the ideal degradable schedule.

In reality, all airlines have itineraries with more than one flight leg; that is, passengers connect through an intermediate or hub airport. Thus, each flight leg will have passengers who are traveling on different itineraries. The importance of a flight leg will not be fully correlated with the importance of any one itinerary alone. In deciding the layer in which to put a flight leg, it is necessary to consider the importance of all the itineraries that use that flight leg.

Because the size of a layer is determined by the number of flight legs in that layer, the number and importance of itineraries in each layer will not be equal to the number and importance of itineraries in the ideal schedule. Given, also, the very practical consideration that a single flight leg might be carrying passengers on a hundred itineraries, the problem becomes a multi-attribute set partitioning problem, as we are seeking to assign the desired number of flight legs to each layer while simultaneously assigning itineraries to layers in order of importance.

This class of problem is often solved by trading off one attribute against the other. However, in this case, the number of flight legs in each layer is of paramount importance because (a) the number of flight legs in the protected layer cannot exceed the maximum number of flight legs that can be operated in bad weather, and (b) layers with limited number of flight legs may not have feasible fleet assignment or aircraft routing solutions. Thus, we address this problem by setting a constraint on the maximum size of every layer except the least important layer, and then maximizing the "weighted" importance of the itineraries in the schedule. However, this approach presents us with two challenges. The first is to decide on the appropriate measure of importance. The second is to determine the appropriate weighting scheme for itineraries as a function of their layer.

There are many possible measures of importance. One can imagine the importance of an itinerary being a function of the number of frequent flyers who use that itinerary. One can also imagine that the revenue an itinerary provides to the airline is an appropriate measure of importance. If we assume that revenue is an indication of importance, the multi-attribute set partitioning problem becomes the problem of assigning the desired number of flight legs to each layer while simultaneously assigning itineraries to layers in order of revenue. The ideal solution would have the highest revenue itineraries in the first or most important layer, the next highest revenue itineraries in the second layer, and so on. Given the considerations described above, our objective is to maximize the weighted revenue of the itineraries in the schedule.

While it is possible to develop a myriad of different weighting schemes, the most important property of any scheme is that the weighting increases monotonically as the importance of the layer increases. Thus, for simplicity, we chose the rank of the layer as the basis for our weighting scheme. Specifically, we decided that the revenue of an itinerary that is in the most important layer would get full weight because this itinerary would be protected from capacity reductions, while the revenue of an itinerary in the least important layer would be given significantly less weight, if any, because this itinerary would be one of the first to be delayed or cancelled when the weather deteriorates. The revenue of an itinerary in a layer between the most and least important layers is given a weight that is proportional to the position of its layer in the ranked order of layers because the itinerary would be partially protected from capacity reductions. The level of protection is related to the position of the layer relative to the most and least important layers. Thus, in the remainder of the paper, the term "protected revenue" is used to describe the weighted revenue of the itineraries in the schedule.

As presented in Chapter 1, there are three different ways to incorporate the degradability into the scheduling process. The integer programming models for each case are presented below. The first model, the degradable schedule partitioning model (D-SPM), is designed to solve the degradable airline scheduling problem before solving the fleet assignment and aircraft routing problems. The second model, the degradable fleet assignment model (D-FAM), is designed to simultaneously solve the degradable airline scheduling problem and the fleet assignment problem. The third model, the degradable aircraft routing model (D-ARM), is designed to simultaneously solve the degradable airline scheduling problem and the aircraft routing problem.

2.1 Degradable Schedule Partitioning Model

The degradable schedule partitioning model is developed using a daily time-line network in which the time and the airport is modeled as a node, each flight leg is modeled as a directed "flight arc" between nodes, and the time that an aircraft spends at an airport is modeled as a directed "ground arc." Because the degradable schedule partitioning model is designed to solve the degradable airline scheduling problem *before* solving the fleet assignment and aircraft routing problems, we do not have to consider fleet assignment costs or aircraft maintenance constraints. We recognize that this may lead to sub-optimal or even infeasible fleet assignments and aircraft routings, but we believe that in a very large airline, where the numbers of flight legs and aircraft are numerous, the loss in optimality, in terms of the estimated profit after planning, will be small relative to the gain in robustness and thus the reduction in recovery costs during operation.

We define sets, decision variables, parameters, indicator variables as follows:

Sets

N = the set of nodes, representing times and airports, indexed by n, m

F = the set of flight arcs indexed by nm

G = the set of ground arcs indexed by nm

A = the set of arcs, $A = F \cup G$

O = the set of overnight arcs indexed by nm, $O \subset A$

I = the set of itineraries indexed by i

K = the set of layers, $\{1 \cdots \bar{K}\}$, indexed by k

Decision Variables

x^k_{nm} = 1, if arc nm is in layer k $\forall nm \in F$

0, otherwise $\forall nm \in F$

the number of airplanes on the ground on arc nm is in layer k $\forall nm \in G$

z^k_i = 1, if the itinerary i is in layer $\bar{k}$ and $\bar{k} \geq k$

0, otherwise

Indicator Variable

δ^i_{nm} = 1, if flight arc nm is in itinerary i

0, otherwise

Parameters

W = number of aircraft available

$\bar{K}$ = number of total layers

$\hat{K}$ = number of protected layers, $\hat{K} < \bar{K}$

v_i^k = revenue of itinerary i, if itinerary is in layer 1,

revenue 'lost' when itinerary i is moved from layer k to $k+1$

$\sum_{k<=k^1} v_i^k > \sum_{k<=k^2} v_i^k \ \forall k$ and $\forall k^1 < k^2$

C_h = number of operations scheduled at hub h per a day

θ_h = maximum percentage of capacity reduction in bad weather at hub h

S^k = number of flights allowed in layer k

$\sum k \leq \hat{K}$ and $S^k \leq \min(1-\theta_h)$

Three things are important to note: First, the definition of z_i^k is similar to the approach taken by Bertsimas and Stock Patterson [13] to deal with problems where there is a potential for fractional solutions. In their paper, they define the binary decision variable as whether the flight arrives at the sector by time t, instead of at time t. In this way, the decision variable has value zero until $t-1$ and value one from t, whereas it has only one t that has value one in the latter case. In the D-SPM model, the decision variable values are reversed. The decision variable has value one for $\bar{k} \leq k$ and has value zero for $\bar{k} > k$. For example, if itinerary i is in layer 3, z_i^1, z_i^2, z_i^3 will have a value of one and other variables for i will be zero.

Second, because layer 1 is the most important layer in the network, all the itineraries in layer 1 are "protected" because that layer is operable in bad weather. For a schedule with two layers, all the revenue of an itinerary is "lost" when that itinerary is moved from layer 1, the protected layer, to layer 2, the unprotected layer. For a schedule with many layers, the revenue that is lost when an itinerary is moved from layer k to layer $k+1$ is a fraction of the total revenue for that itinerary, as the likelihood that a flight leg in layer k is delayed or cancelled is lower than the likelihood that a flight leg in layer $k+1$ is delayed or cancelled.

Third, C_h is the number of operations scheduled at hub h per a day and θ_h is the maximum capacity reduction in bad weather at hub h from historical data. Thus $C_h(1-\theta_h)$ gives the minimum number of operations an airline can have at hub h even in bad weather.

Given the definitions above, the degradable schedule partitioning model (D-SPM) can be formulated as

$$\max \quad \sum_i \sum_k v_i^k z_i^k \tag{2.1}$$

$$\text{s.t.} \quad \sum_{nm \in F} x_{nm}^k \le S^k \quad \forall k < \bar{K} \tag{2.2}$$

$$\sum_k x_{nm}^k = 1 \quad \forall nm \in F \tag{2.3}$$

$$\sum_m x_{nm}^k - \sum_m x_{mn}^k = 0 \quad \forall n \in N,\ \forall k \in K \tag{2.4}$$

$$\sum_{k \le \hat{K}} \sum_{nm \in F} \sum_{n \ni h} x_{nm}^k \le C_h(1 - \theta_h) \quad \forall h \in H \tag{2.5}$$

$$\sum_{k \in K} \sum_{nm \in O} x_{nm}^k \le W \tag{2.6}$$

$$z_i^k \ge x_{nm}^k \delta_{nm}^i \quad \forall nm \in F,\ \forall i \in I,\ \forall k \in K \tag{2.7}$$

$$z_i^k \ge z_i^{k+1} \quad \forall k < \bar{K} \tag{2.8}$$

$$x_{nm}^k \in Z^+ \quad \forall nm \in G,\ \forall k \in K \tag{2.9}$$

$$x_{nm}^k \in \{0,1\} \quad \forall nm \in F,\ \forall k \in K \tag{2.10}$$

$$z_i^k \in \{0,1\} \quad \forall i \in I,\ \forall k \in K \tag{2.11}$$

The objective function (2.1) maximizes the protected revenue. The constraints (2.2) ensure that the number of flight legs in each layer does not exceed the desired number of flight legs, S^k, in that layer. The constraints (2.3) are cover constraints ensuring that each flight leg is in exactly one layer. The constraint (2.4) is a conservation of flow constraint, which ensures that the number of aircraft at an airport does not increase or decrease over time. The constraints (2.5) ensure that, at each hub airport, the number of operations in protected layers is less than the capacity of the hub airport in bad weather. The constraint (2.6) is a count constraint ensuring that the number of aircraft used in the model is less than or equal to the number of available aircraft. The constraints (2.7) and (2.8) are used to identify the layer in which an itinerary is located. The constraint (2.7) forces z_i^k to have a value of one if any of its constituent flight legs are in layer k. The constraint (2.8) makes z a step variable. For example, if there are 4 layers and an itinerary i has two flight legs, one in layer 1 and one in layer 3, from (2.7), z_i^1 and z_i^3 will have value one. In that case, (2.8) forces z_i^2

to be one as well. In fact there is no constraint for z_i^4; however, from the nature of the objective function, v_i^k always has negative value for $k > 1$, and it will make the z_i^4 value zero. In this way, we consider the itinerary to be in 'the least important layer' of its constituent flight legs when they are not assigned to the same layer. The constraints (2.9), (2.10), and (2.11) are for integrality of decision variables. The decision variables for flight legs are binary variables, and the decision variables for ground arcs can have any non-negative integer value.

2.2 Degradable Fleet Assignment Model

The degradable fleet assignment model is also developed using a daily time-line network. The major difference and the challenge of D-FAM is that we combine two optimizations – minimizing fleet assignment cost and maximizing degradable airline scheduling value. Although both optimizations use dollars as the measure, they cannot be combined because the fleet assignment value is an actual cost to the airline while the degradable airline scheduling value is merely a value that is proportional to the revenue of the itineraries an airline would protect by having a degradable airline schedule, which is not the actual revenue or profit an airline makes.

When there are two objective functions, $f_1(x)$ and $f_2(x)$, one way to handle them is to simply combine them as a single objective function. This approach works only when $f_1(x)$ and $f_2(x)$ have the same measure with the same scale. If two functions use the same measure but different scale or different measure which is convertible to each other, we can use the weighted sum of two functions, $f_1(x) + \alpha f_2(x)$, instead. However, the question remains as to how to pick the right value of α.

Another technique is to put one of the functions as a constraint to optimize another function. First, we optimize over $f_1(x)$ only, and get the optimal value $f_1^*(x)$. Then we optimize over $f_2(x)$ with an additional constraint, which is $f_1(x)$. The value for $f_1(x)$ is bounded by $(1+\epsilon)f_1^*(x)$, where $\epsilon > 0$. After we optimize over $f_2(x)$ for various permutation values for $f_1^*(x)$, we can have a graph showing the trade-off between

$f_1(x)$ and $f_2(x)$.

In D-FAM, we use the latter approach. D-FAM is solved in two stages. In the first stage, the Fleet Assignment Model (FAM) is solved to obtain the minimum fleet assignment cost, Z^*, without degradable airline scheduling constraints. In the second stage, we use the same objective function as D-SPM to maximize the protected revenue, as well as all constraints for FAM and D-SPM. In addition, we have one more constraint for the objective function value of FAM, the fleet assignment cost. In that additional constraint, some positive increase of the minimum FAM value from the first stage is used as an upper boundary for FAM cost for D-FAM. Details for each stage are described in the following subsections.

2.2.1 Basic Fleet Assignment Model

The Basic Fleet Assignment Model consists of the objective function that minimizes the fleet assignment cost such that 1) all flights are flown by exactly one fleet type, 2) the aircraft flow is balanced, and 3) only the number of available aircraft is used. Or, mathematically, FAM can be formulated as follows:

Sets

N = the set of nodes, representing times and airports, indexed by n, m

F = the set of flight arcs indexed by nm

G = the set of ground arcs indexed by nm

A = the set of arcs, $A = F \cup G$

O = the set of overnight arcs indexed by nm, $O \subset A$

P = the set of different fleet types indexed by p

Decision Variables

$x_{nm,p}$ = 1, if arc $nm \in A$ is assigned to fleet type $p \in P$ $\quad \forall nm \in F$

0, otherwise $\quad \forall nm \in F$

number of airplanes on the ground on arc nm $\quad \forall nm \in G$

Parameters

$c_{nm,p}$ = cost of flight nm if fleet type p is assigned

0, if $nm \in G$

W_p = the number of aircraft in fleet type p, $\forall p \in P$

$$\min \quad \sum_{nm \in F} \sum_{p \in P} c_{nm,p} x_{nm,p} \tag{2.12}$$

$$\text{s.t.} \quad \sum_{p \in P} x_{nm,p} = 1 \quad \forall nm\ inF \tag{2.13}$$

$$\sum_{m \in N} x_{nm}^{k} - \sum_{m \in N} x_{mn}^{k} = 0 \quad \forall n \in N,\ \forall p \in P \tag{2.14}$$

$$\sum_{nm \in O} x_{nm,p} \leq W_p \quad \forall p \in P \tag{2.15}$$

$$x_{nm,p} \in Z^{+} \quad \forall nm \in G,\ \forall p \in P \tag{2.16}$$

$$x_{nm,p} \in \{0, 1\} \quad \forall nm \in F,\ \forall p \in P \tag{2.17}$$

The constraints (2.13) are cover constraints ensuring that each flight leg is covered once and only once by a fleet type. Constraints (2.14) are conservation of flow constraints ensuring the number of aircraft coming in to each airport is the same as the number of aircraft going out of the airport. Constraints (2.15) are count constraints ensuring that only the available number of aircraft of each fleet type is used in the assignment. The constraints (2.16) and (2.17) are for integrality of decision variables. The objective function (2.12) is to minimize the total cost.

The objective function coefficient $c_{nm,p}$ is the summation of all costs of flight nm flown by fleet type p. The cost includes operating cost and spill cost. Variations on fleet assignment models and approaches were introduced by Hane et. al [36], Rushmeier and Kontogiorgis [49], Clarke et. al [19], Berge and Hopperstad [12], Kontogiorgis and Acharya [42], and Abara [2]. Barnhart et al. [10] showed that the fleet assignment model can be improved even more using an itinerary-based model.

In our model, we use one of the earliest and simplest models. The cost is calculated as the sum of the operating cost, which includes fuel, maintenance, crew cost for each fleet type, and spill cost, which is the summation of fares for passengers who cannot get served because of limited fleet capacity. We assume that there is no recaptured revenue, and each flight leg in an itinerary has the value of the whole itinerary.

2.2.2 Multi-criteria Optimization

In D-FAM, we not only have two objective functions; we also have two decisions to make in selecting a fleet type and layer. Once we solve a FAM, as described in the previous section, we have the minimum FAM cost, Z^*. The objective function from the first stage, Z^*, is an input for D-FAM in the second stage. The D-FAM formulation for maximizing the protected revenue while assigning a fleet type and layer to each flight and the FAM cost is not greater than the given permutation of Z^*. D-FAM is basically a combination of FAM and D-SPM. The objective function is the same as D-SPM. The constraints consist of all constraints for fleet assignment feasibility and degradable airline scheduling feasibility. In addition, there is a constraint for FAM cost. We define the following:

Sets

N = the set of nodes, representing times and airports, indexed by n, m

F = the set of flight arcs indexed by nm

G = the set of ground arcs indexed by nm

A = the set of arcs, $A = F \cup G$

O = the set of overnight arcs indexed by nm, $O \subset A$

P = the set of different fleet types indexed by p

I = the set of itineraries indexed by i

K = the set of layers indexed by k

Decision Variables

$x^k_{nm,p}$ = 1, if arc nm is assigned to fleet p and in layer k
0, otherwise

z^k_i = 1, if itinerary i is in layer $\bar{k}$ and $\bar{k} \geq k$
0, otherwise

Indicator Variable

δ^i_{nm} = 1, if flight arc nm is in itinerary i

0, otherwise

Parameters

$c_{nm,p}$ = cost of flight nm if fleet type p is assigned

0, if $nm \in G$

W_p = the number of aircraft in fleet type p, $\forall p \in P$

$\bar{K}$ = number of total layers

$\hat{K}$ = number of protected layers, $\hat{K} < \bar{K}$

v_i^k = revenue of itinerary i, if itinerary is in layer 1,

revenue 'lost' when itinerary i is moved from layer k to $k+1$

$\sum_{k<=k^1} v_i^k > \sum_{k<=k^2} v_i^k \forall k$ and $\forall k^1 < k^2$

C_h = number of operations scheduled at hub h per a day

θ_h = maximum percentage of capacity reduction in bad weather at hub h

S^k = number of flights allowed in layer k

$\sum k \leq \hat{K}$ and $S^k \leq \min(1-\theta_h)$

Z^* = the minimum FAM cost without considering degradable airline scheduling obtained from a simple FAM in the first stage

ϵ = permutation parameter for Z^*

The only difference in the sets between D-SPM and D-FAM is P, the set of different fleet types. The decision variable z is the same as in D-SPM. The decision variable x is defined simply by combining the decision variables in D-SPM, x^k_{nm}, and the decision variables in FAM, $x_{nm,p}$. Given the definitions above, the degradable schedule partitioning model (D-FAM) can be formulated as

$$\max \quad \sum_i \sum_k v_i^k z_i^k \tag{2.18}$$

$$\text{s.t.} \quad \sum_{p \in P} \sum_{nm \in F} x^k_{nm,p} \leq S^k \quad \forall k < \bar{K} \tag{2.19}$$

$$\sum_p \sum_k x^k_{nm} = 1 \quad \forall nm \in F \tag{2.20}$$

$$\sum_m x^k_{nm,p} - \sum_m x^k_{mn,p} = 0 \quad \forall n \in N, \forall k \in K, \forall p \in P \tag{2.21}$$

$$\sum_{k \leq \bar{K}} \sum_{p \in P} \sum_{nm \in F, n \ni h} x^k_{nm,p} \leq C_h(1 - \theta_h) \quad \forall h \in H \tag{2.22}$$

$$\sum_{k \in K} \sum_{nm \in O} x^k_{nm,p} \leq W_p \quad \forall p \in P \tag{2.23}$$

$$z^k_i \geq \sum_{p \in P} x^k_{nm,p} \delta^i_{nm} \quad \forall nm \in F, \forall i \in I, \forall k \in K \tag{2.24}$$

$$z^k_i \geq z^{k+1}_i \quad \forall k < \bar{K} \tag{2.25}$$

$$\sum_{nm \in F} \sum_{p \in P} \sum_{k \in K} c_{nm,p} x^k_{nm,p} \leq Z^*(1 + \epsilon) \tag{2.26}$$

$$x^k_{nm,p} \in Z^+ \quad \forall nm \in G, \forall k \in K, \forall p \in P \tag{2.27}$$

$$x^k_{nm,p} \in \{0, 1\} \quad \forall nm \in F, \forall k \in K, \forall p \in P \tag{2.28}$$

$$z^k_i \in \{0, 1\} \quad \forall i \in I, \forall k \in K \tag{2.29}$$

The objective function (2.18) is exactly the same as D-SPM. The constraints (2.19) and (2.22) ensure degradable scheduling feasibility. That is, they ensure that the number of flight legs in each layer does not exceed the desired number of flight legs in each layer regardless of its fleet type, and the number of operations at the hub airports in protected layers does not exceed the airport capacity in bad weather at the hub. The constraints (2.20), (2.21), and (2.23) ensure both FAM and D-SPM feasibility. The constraint (2.20) is a cover constraint, now defined for both fleet type and layer. The constraints (2.21) are the conservation of flow constraint for each fleet type and each layer. The constraints (2.23) are counting constraints for the number of aircraft used for each fleet type. The constraints (2.24) and (2.25) are used to identify the layer in which an itinerary is located. They are the same as D-SPM as well. The constraint (2.26) links the trade-off between FAM cost and D-SPM value. For given ϵ, D-FAM will decide the optimal fleet and layer assignment to maximize the total protected revenue. The constraint (2.27), (2.28), (2.29) ensure the integrality of the decision variables.

When D-FAM is solved for different ϵ values, a graph of the results describes the trade-off between FAM cost and DAS value.

2.3 Degradable Aircraft Routing Model

The degradable aircraft routing model (D-ARM) has major differences from D-SPM or D-FAM. First, we use entire aircraft maintenance routings or 'routes' as decision variables instead of individual flight legs. Second, D-ARM is developed using a 3-day-long time-line network, instead of a daily network.

A route is defined as a sequence of flight legs which lasts for three days, and satisfies flow balance and maintenance requirements. By requiring that each route stops at a maintenance station at least once in three days, we guarantee the maintenance for every aircraft at least once in every five days. The routing variables can be classified as composite variables, introduced by Cohn [20], which pass some of the problem complexity to the sub-problem. The flight string model by Barnhart et al. [7] has a similar idea. The string is defined as a sequence of flight legs that start and end at (possibly different) maintenance stations, satisfy flow balance, and are maintenance feasible.

We define the routes using the 3-day long network instead of the daily network. Each day has the same schedule. However, we have dated operations. In a dated operation, we can consider maintenance opportunities for individual aircraft. Each aircraft does not have to repeat the same route which provides more flexibility for routings. This approach also allows us to apply revenue of different days of the week, instead of using the average revenue. However, D-ARM does not necessarily capture all routings in the string model, and vice versa.

We define the following:

Sets

N = the set of nodes, representing times and airports, indexed by n, m

F = the set of flight arcs indexed by nm

G = the set of ground arcs indexed by nm

A = the set of arcs, $A = F \cup G$

O = the set of overnight arcs indexed by nm, $O \subset A$

I = the set of itineraries indexed by i

P = the set of fleet types indexed by p

K = the set of layers, $\{1 \cdots \bar{K}\}$, indexed by k

R = the set of routes indexed by r

D = the set of days, $\{1, 2, 3\}$, indexed by d

Decision Variables

y_r^k = 1, if route r is in layer k

0, otherwise

z_i^k = 1, if the itinerary i is in layer $\bar{k}$ and $\bar{k} \geq k$

0, otherwise

Indicator Variable

δ_{nm}^i = 1, if flight arc nm is in itinerary i

0, otherwise

γ_{nm}^d = 1, if flight arc nm is in day d

0, otherwise

λ_r^d = 1, if fleet type p is used for route r

0, otherwise

Parameters

W_p = number of aircraft available in fleet type p, $\forall p \in P$

α_r^d = number of flights in the route r in day d

$\alpha_r^{d,h}$ = number of flights in the route r which depart from hub h in day d

v_i^k = revenue of itinerary i, if itinerary is in layer 1,

revenue 'lost' when itinerary i is moved from layer k to $k+1$

$\sum_{k<=k^1} v_i^k > \sum_{k<=k^2} v_i^k \forall k$ and $\forall k^1 < k^2$

C_h = number of operations scheduled at hub h per a day

θ_h = maximum percentage of capacity reduction in bad weather at hub h

S^k = number of flights allowed in layer k

$\sum k \leq \hat{K}$and $S^k \leq \min(1 - \theta_h)$

Given the definitions above, the degradable aircraft routing model (D-ARM) can be formulated as:

$$\max \quad \sum_k \sum_i v_i^k z_i^k \tag{2.30}$$

$$\text{s.t.} \quad \sum_r \alpha_r^d y_r^k \leq S^k \quad \forall k < \bar{K} \; \forall d \in D \tag{2.31}$$

$$\sum_{k, r \ni nm} y_r^k = 1 \quad \forall nm \in F \tag{2.32}$$

$$\sum_{k \leq \hat{K}, r} \alpha_r^{d,h} y_r^k \leq C_h (1 - \theta_h) \quad \forall h \in H, \; \forall d \in D \tag{2.33}$$

$$\sum_{k \in K} \sum_r y_r^k \lambda_r^d \leq W_p \quad \forall p \in P \tag{2.34}$$

$$z_i^k \geq \sum_{r \ni nm} y_r^k \delta_{nm}^i \quad \forall nm \in F, \; \forall i, \; \forall k \tag{2.35}$$

$$z_i^k \geq z_i^{k+1} \quad \forall k < \bar{K} \tag{2.36}$$

$$y_r^k \in \{0, 1\} \quad \forall r, \; \forall k \tag{2.37}$$

$$z_i^k \in \{0, 1\} \quad \forall i, \; \forall k \tag{2.38}$$

The objective function (2.30) maximizes the protected revenue. The constraints (2.31) ensure that the number of flight legs in each layer does not exceed the desired number of flight legs, S^k, in that layer for each day. The constraints (2.32) ensure that each flight leg is in exactly one layer. The constraints (2.33) ensure that at each hub airport, the number of operations in protected layers is less than the capacity of the hub airport in bad weather for each day. The constraints (2.34) ensure that the number of aircraft used in the model is less than or equal to the number of available aircraft. The constraints (2.35) and (2.36) are used to identify the layer in which an itinerary is located. The constraints (2.37) and (2.38) ensure that decision variables are binary.

Chapter 3

Implementation

To illustrate how the idea of a degradable airline scheduling might be implemented in an airline, we developed a two layer degradable schedule for a *representative* airline using each model we developed in Chapter 2. In the following sections, we introduce the characteristics of the representative airline including its flight schedule, passengers, itineraries, revenue, and explain how the model parameters for the representative airline were derived. We also explain how model tractability was achieved by reducing the number of decision variables without compromizing on fidelity.

3.1 Prototype Example

The schedule of the representative airline is patterned on the domestic schedule of a major U.S. airline. The airline in question has three U.S. hub airports: one in the East, one in the Mid-West, and one in the South. During the time period for which we had the requisite data, the airline was serving 76 domestic destinations with 265 aircraft (of seven fleet types) and 1,134 flights. On an average day during that time period, the airline received approximately \$12,912,229 in revenue from 66,643 passengers, flying 20,655 itineraries. Of the 20,655 itineraries, 57.8 percent had only a single flight leg, 41.7 percent had two flight legs, and only 0.5 percent had three flight legs.

There are several important points worth noting about this representative airline. First, its flight schedule is the same as the domestic flight schedule of the major U.S. airline on which the representative airline is patterned. Thus, the departure and arrival times for all flight legs are realistic. Second, the representative airline has a very clear hub-and-spoke network structure. Only six flight legs out of 1,134 are from non-hub airport to non-hub airport. As will be shown, this unique characteristic of the airline allows us to develop heuristics for D-ARM. Third, the routings of the representative airline were developed using the baseline FAM, and an aircraft maintenance routing model determined by first-come-first-out basis. These fleet assignment and aircraft routing models are the simpliest method applied by airlines. Thus, the routings in the representative airline are reflective of what an airline of this size might develop. We assume that the revenue and passengers on each itinerary for the representative airline are the same as those for the domestic markets of the major U.S. airline.

3.2 Parameters

There are few parameters in the mathematical formulations of the degradable airline scheduling models. First of all, we need to decide how many layers we want to protect and how many layers we do not want to protect. Given the number of layers, we need to decide the size of each layer, that is, how many flight legs we allow in each layer. To decide the size of layers, especially size for protected layers, the capacity at the hub, and how much it is reduced in bad weather, needs to be calculated. How much revenue is protected if an itinerary is in each layer need to be decided as well.

Number of Layers

As stated above, the prototype degradable schedule has two layers for the following reasons. First, this is the minimum number of layers that a degradable schedule can have, as there must be at least one protected and one unprotected layer. Second, the fewer the number of layers, the greater the size of the layers, and therefore, the greater the likelihood that each layer has a feasible solution for aircraft routing with the same number of aircraft. Since we only have two layers, the protected layer will be labeled as *Layer 1*, and the unprotected layer will be labeled as *Layer 2*.

Hub Airport Capacity

While we can easily figure out the number of operations in good weather by examining the schedule, it is not as easy to determine the number of operations that can be performed when the capacity is reduced due to bad weather at the hub airports. Thus, a more detailed analysis of airports observatoins is required.

The FAA Airport Capacity Benchmark Report [5] documents the observed capacity at the 31 busiest U.S. domestic airports under good and bad weather conditions. Because one of the hub airports was not included in the Airport Capacity Benchmark Report, we took the capacity information for six spoke airports among the top ten airports for the airline in question [1], and used it as an estimation for the capacity reduction rate at the third hub airport in the most conservative way.

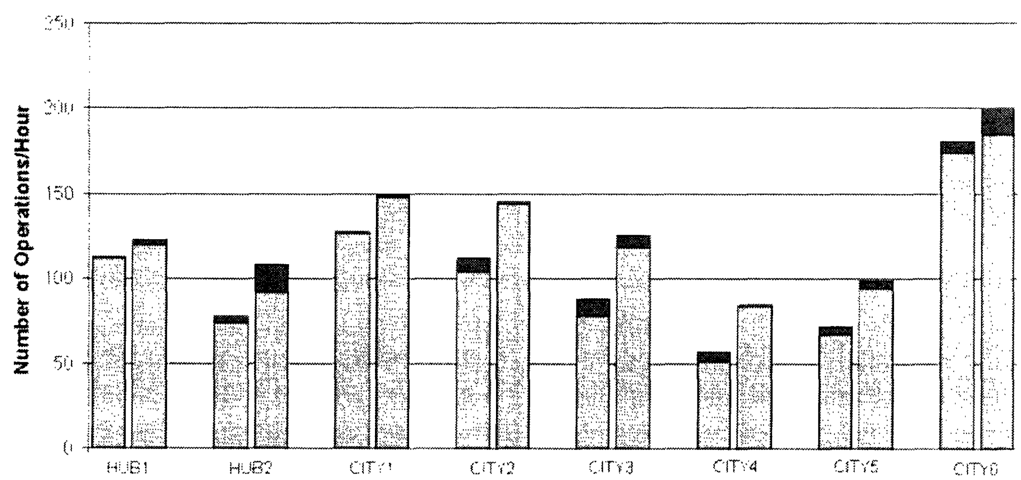

Figure 3-1: Airport Capacity

Figure 3-1 shows the range of capacity for the eight airports considered. In the figure, there are two bars for each airport. The bar on the left shows the capacity of the airport in bad weather. The bar on the right shows the capacity of the airport in normal weather. The darker area on each bar represents the observed variability in capacity under each condition. This variability may be explained by the fact that, while the choice between IFR (applied in bad weather) and VFR (applied in normal weather) is based on ceiling and visibility conditions, airport capacity is also a function of the runway configuration in use, which in turn is a function of the wind direction and speed.

The capacity changes at each airport are summarized in Table 3.1 [24]. As the table shows, the largest capacity reduction for the airports considered is approximately 40 percent. Thus, we assume the capacity reduction for HUB3 as 40 percent.

Table 3.1: Airport Capacity Reduction

Airport	minimum reduction (%)	maximum reduction (%)
HUB1	5.83	8.94
HUB2	15.22	31.48
CITY1	13.51	15.33
CITY2	22.22	28.28
CITY3	25.42	38.10
CITY4	32.14	38.82
CITY5	24.21	32.32
CITY6	5.95	16.50

Size of the Layers

Having determined the number of layers, the next task was to determine the size of each layer. Given our goal of protecting the flight legs in the protected layer during bad weather, the number of flights that can be placed in the protected layer must be less than or equal to the fraction of the bad weather capacity that the airline might anticipate. Because airport capacity is allocated to airlines in proportion to their share of the total number of operations that are nominally scheduled at any given airport, we assume that the fraction of the total capacity available to an airline during bad weather is the same as the fraction of total capacity available to the airline in normal weather. Thus, the change in airport capacity may be used as a proxy for the change in the capacity available to the airline.

A complete treatment of capacity constraints on layer size would nominally require an analysis of the capacity constraints at all the airports an airline serves. However. the hub-and-spoke nature of the airline in question – only 0.5 percent of flights are non-stop flights between spoke airports – results in a situation where its operations are constrained by the capacity at its hub airports. Therefore, instead of considering the capacity constraints at all the airports in the U.S., we need only to consider the capacity constraints at the hub airports. From Table 3.1 and the assumption we made about HUB3. the maximum capacity reduction for each hub is 8.94 percent. 31.48 percent, and 40 percent, respectively. With the most conservative approach, to guarantee that the flight legs in the protected layer can continue to operate when the weather deteriorates, the size of Layer 1 must be no more than 60 percent.

Weighted Revenue

For itineraries in Layer 1, it is easy to assume that the revenue protected is the same as the total revenue for the itinerary since the flight legs in Layer 1 are always protected and have priorities over flight legs in Layer 2. For itineraries in Layer 2, the unprotected layer, the protected revenue could be anything between zero and full revenue. It could be strictly proportional to the revenue, or include other factors such as market share.

In this case, for the sake of simplicity, we take the most conservative view and assume no revenue is protected when the itinerary is not in the protected layer.

3.3 Reducing the Problem Size

The final consideration is the size of the problem, as this determines both computation time and memory requirements. To limit the size of the problem, we only consider itineraries with revenue greater than the average revenue. This decision greatly reduces the number of itineraries that must be considered without sacrificing the fidelity of the solution. To support this point, consider the distribution of revenue for the major U.S. airline on which the representative airline is patterned.

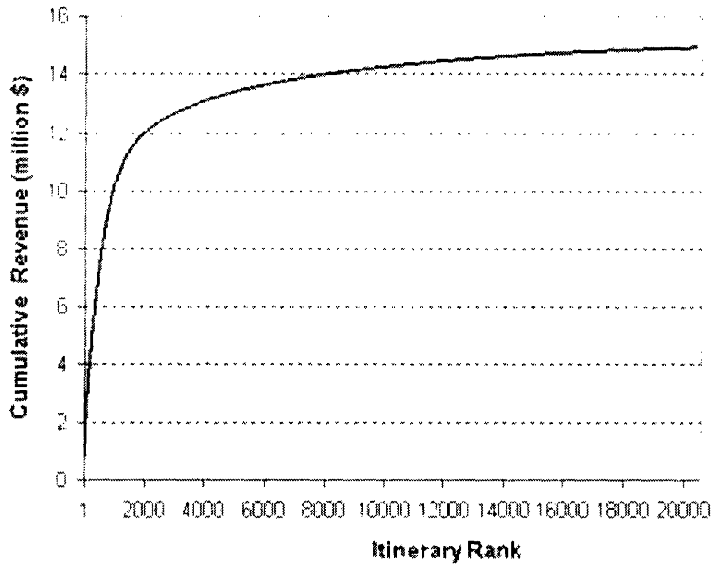

Figure 3-2: Accumulated Itinerary Revenue

Figure 3-2 shows the cumulative average daily revenue for all the itineraries on an average day for the airline in question. As the figure shows, 8.4 percent of the itineraries generate 83.9 percent of the total revenue. In fact, only 1,744 itineraries generate more than the average daily revenue of \$626. Among these 1,744 itineraries, 64.8 percent are single flight leg itineraries, and only one itinerary has more than two flight legs.

3.4 Model Simplification

Since we only have two layers and we assume that the protected revenue is zero if an itinerary is in Layer 2, we can simplify our models for this specific case. This procedure reduces the number of decision variables and the number of constraints, thus, reduces the problem size.

Instead of defining z_i^k, we define z_i as one if itinerary i is in layer 1 and zero otherwise. That means, if z_i is zero, that automatically makes the itinerary i to be in layer 2. The objective function, therefore, can be re-written as follows:

$$\sum_i v_i z_i \tag{3.1}$$

where v_i is the revenue for itinerary i. This is equivalent to the original objective function,

$$\sum_k \sum_i v_i^k z_i^k \tag{3.2}$$

because v_i^2 has value $-v_i^1$. Also, if z_i^2 has value one, z_i^1 also has value one. Thus the contribution to the objective function is zero. The problem constraints where we identify the layer in which an itinerary is located can be simplified as well. There are some variations depending on which model we use. D-SPM, D-FAM, D-ARM, have a similar constraints structure. In D-SPM, we have:

$$z_i^k \geq x_{nm}^k \delta_{nm}^i \quad \forall nm \in F\ \forall i \in I,\ \forall k \in K \tag{3.3}$$

$$z_i^k \geq z_i^{k+1} \quad \forall k < \hat{K}. \tag{3.4}$$

The constraint (3.4) is common in all three IP formulations: constraint (2.8) in D-SPM, constraint (2.25) in D-FAM, and constraint (2.36) in D-ARM. Although the right hand side of the constraint (3.3) is different depending on how we define the decision variable, the constraints (2.7), (2.24), and (2.35) all make z_i^k value one, if any of its flight legs in itinerary i is in layer k. Instead, we can simplify them as follows:

$$z_i \leq x_{nm}^1 \delta_{nm}^i \quad \forall nm \in F\ \forall i \in I. \tag{3.5}$$

In constraints (3.5), the right hand side value is one if the flight leg is in the Layer 1 and zero if it is in the Layer 2 by the cover constraints. Since this is a maximization problem and v_i has a positive value, z_i would have a value of one if there are no constraints on it. The constraint (3.5) forces z_i value to zero, if any of its flight legs are in Layer 2.

For D-ARM, we can simplify those constraints as follows:

$$z_i \leq \sum_{r \ni nm} y_r^1 \delta_{nm}^f \quad \forall nm \in F \, \forall i \in I. \tag{3.6}$$

In summary, the simplied formulations are as follows:

(D-SPM)

$$\max \quad \sum_i v_i z_i \tag{3.7}$$

$$\text{s.t.} \quad \sum_{nm \in F} x_{nm}^1 \leq S^1 \tag{3.8}$$

$$\sum_k x_{nm}^k = 1 \quad \forall nm \in F \tag{3.9}$$

$$\sum_m x_{nm}^k - \sum_m x_{mn}^k = 0 \quad \forall n \in N, \, \forall k \in K \tag{3.10}$$

$$\sum_{nm \in F} \sum_{n \ni h} x_{nm}^1 \leq C_h(1 - \theta_h) \quad \forall h \in H \tag{3.11}$$

$$\sum_{k \in K} \sum_{nm \in O} x_{nm}^k \leq W \tag{3.12}$$

$$z_i \leq x_{nm}^1 \delta_{nm}^i \quad \forall nm \in F \, \forall i \tag{3.13}$$

$$x_{nm}^k \in Z^+ \quad \forall nm \in G, \, \forall k \in K \tag{3.14}$$

$$x_{nm}^k \in \{0,1\} \quad \forall nm \in F, \, \forall k \in \tag{3.15}$$

$$z_i \in \{0,1\} \quad \forall i \in I \tag{3.16}$$

(D-FAM)

$$\max \quad \sum_i v_i z_i \tag{3.17}$$

$$\text{s.t.} \quad \sum_{p \in P} \sum_{nm \in F} x_{nm,p}^1 \leq S^1 \tag{3.18}$$

$$\sum_p \sum_k x_{nm}^k = 1 \quad \forall nm \in F \tag{3.19}$$

$$\sum_m x_{nm,p}^k - \sum_m x_{mn,p}^k = 0 \quad \forall n \in N, \, \forall k \in K, \, \forall p \in P \tag{3.20}$$

$$\sum_{p \in P} \sum_{nm \in F, n \ni h} x^1_{nm,p} \leq C_h(1 - \theta_h) \quad \forall h \in H \tag{3.21}$$

$$\sum_{k \in K} \sum_{nm \in O} x^k_{nm,p} \leq W_p \quad \forall p \in P \tag{3.22}$$

$$z_i \leq \sum_{p \in P} x^1_{nm,p} \delta^i_{nm} \quad \forall nm \in F \, \forall i \tag{3.23}$$

$$\sum_{nm \in F} \sum_{p \in P} \sum_{k \in K} c_{nm,p} x^k_{nm,p} \leq Z^*(1 + \epsilon) \tag{3.24}$$

$$x^k_{nm,p} \in Z^+ \quad \forall nm \in G, \, \forall k \in K, \, \forall p \in P \tag{3.25}$$

$$x^k_{nm,p} \in \{0, 1\} \quad \forall nm \in F, \, \forall k \in K, \, \forall p \in P \tag{3.26}$$

$$z_i \in \{0, 1\} \quad \forall i \in I \tag{3.27}$$

(D-ARM)

$$\max \quad \sum_i v_i z_i \tag{3.28}$$

$$\text{s.t.} \quad \sum_r \alpha^d_r y^1_r \leq S^1 \quad \forall d \in D \tag{3.29}$$

$$\sum_{k, r \ni nm} y^k_r = 1 \quad \forall nm \in F \tag{3.30}$$

$$\sum_r \alpha^{d,h}_r y^1_r \leq C_h(1 - \theta_h) \quad \forall h \in H, \, \forall d \in D \tag{3.31}$$

$$\sum_{k \in K} \sum_r y^k_r \lambda^d_r \leq W_p \quad \forall p \in P \tag{3.32}$$

$$z_i \leq \sum_{r \ni nm} y^1_r \delta^f_{nm} \quad \forall nm \in F \, \forall i \tag{3.33}$$

$$y^k_r \in \{0, 1\} \quad \forall r, \, \forall k \tag{3.34}$$

$$z_i \in \{0, 1\} \quad \forall i \tag{3.35}$$

Chapter 4

Solution Approaches

In this chapter, solution approaches for D-SPM, D-FAM and D-ARM are presented. Each model is solved with parameters obtained in Chapter 3, but the approaches to the solutions presented in this chapter can be applied to problems with different parameter values.

Based on the layer assignment, we investigated the distribution of itineraries, revenue, and passengers in these schedules, relative to the nominal schedule for that airline. Via simulation, we also determined the robustness and delay characteristics of the degradable schedule, as well as passenger service quality and recovery cost.

D-SPM has a relatively small problem size compared to D-FAM or D-ARM. D-SPM is directly fed into CPLEX 7.0, using a workstation with 2 CPUs, a 2.4 GHz processor speed, 1 GB RAM, and 512 KB cache. Branch-and-bound is used as the CPLEX default setting. Gomory cuts are applied in this setting. The up branch first strategy as used as a branch-and-bound strategy.

4.1 D-FAM: Multi-criteria Optimization

The solution approach for D-FAM can be summarized as follows:

BEGIN

Solve FAM. Let Z^* be the optimal objective function value.

for each $\epsilon_n > 0$ **do**

Set the constraint: FAM value $\leq Z^*(1 + \epsilon_n)$

Solve D-FAM

Set Z_n = FAM value, X_n = optimal objective function value for D-FAM

end for

Plot a graph for (Z_n, X_n)

END

For each ϵ_n, the D-FAM is fed into CPLEX. D-FAM has a similar constraint structure to D-SPM. However, due to a much bigger problem size, at times we only have a feasible solution one to three percent within an optimal solution. The resulting graph of the trade-off between daily fleet assignment cost and DAS value is shown in Figure 4-1.

Note that the vertical dotted line on the graph is the Z^* value. Since we set $\epsilon > 0$, all our solutions have a higher FAM cost. The horizontal dotted line is the maximum DAS value that is the optimal objective function value for D-SPM. The point near the top right on the dotted line represents the solution with optimal DAS value and feasible fleet assignment. However, the FAM cost is significantly higher than the original FAM cost, Z^*.

We can observe the trade-off between FAM cost and DAS value. In general, higher FAM cost generates higher DAS value solution. However, the marginal increase in DAS value reduces dramatically as the FAM cost gets higher. The point that gives the highest marginal improvement per FAM cost is the one on the upper left. Rather than use this point, we picked the (*) point in Figure 4-1, where ϵ is 0.0016. Although we optimize two objective functions, both of them cannot be treated the same. The FAM cost is the actual cost an airline has to spend where as the DAS value is the total protected revenue that may not bring the actual dollars to an airline. For that reason, we favored points with lower FAM cost.

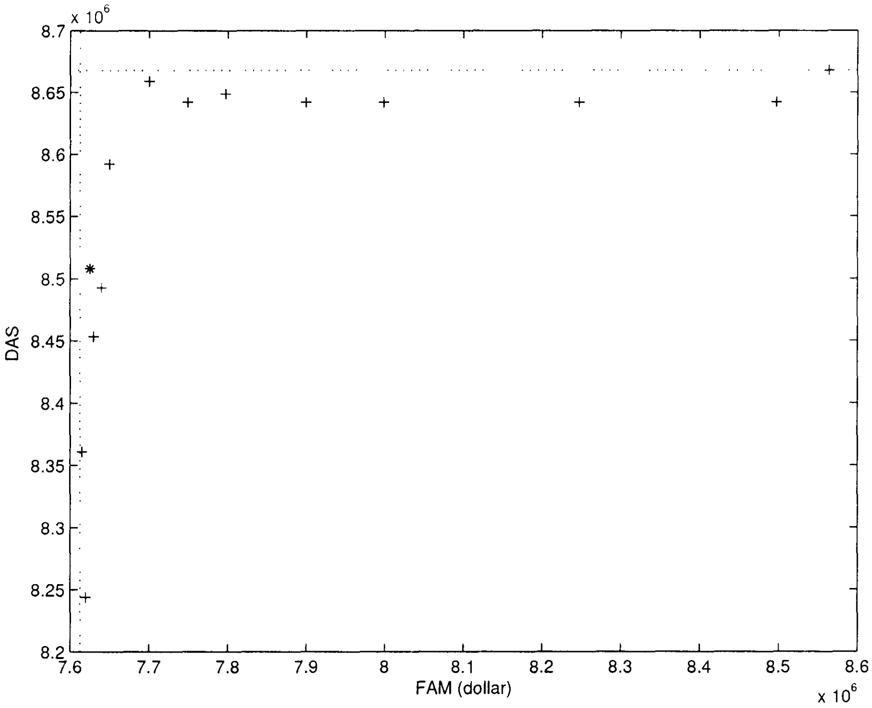

Figure 4-1: Trade-off between FAM and DAS in D-FAM

4.2 D-ARM: Search Heuristics

Because the number of possible routes in D-ARM is very large, it is not feasible to enumerate all the possible routes as variables. Barnhart et al. [7] defined the string as a decision variable. They used column generation to solve the problem. Barnhart et al. [8][9] also introduced branch-and-price and branch-and-price-and-cut algorithms to solve huge integer programs. Cordeau et al. [22] used column generation with benders decomposition to solve aircraft routing and crew scheduling. Desaulniers et. al [23] used column generation, and Dantzig-Wolfe decomposition to solve the linear relaxation of daily aircraft routing where they used branch-and-bound. Ioachim et. al [37] also used the column generation and dynamic programming for the sub-problem. Klabjan et al. [40][39] solved airline crew scheduling by a column generation type approach. The authors had a similar obstacle in the aircraft routing, which is the number of decision variables. They add more variables from random generations.

Although there are several variations in the methodologies of the above authors, the general idea is to start with a subset of variables and add more until we get a near optimal solution.

We introduce search algorithms for aircraft routing. The search algorithm has following three steps. First, we get a set of routes that covers the schedule, as an initial set. Then, we solve D-ARM with the set of routes. Because the given set is a feasible routing, a feasible solution is guaranteed. This second step assigns routes to layers and gives us an initial feasible solution to D-ARM. From the initial feasible solution, a search algorithm is applied to improve a solution.

In this section, we introduce the greedy flight leg pairing algorithm that is used to generate an initial set of routes. We then solve D-ARM to get an initial feasible solution to D-ARM, which is then used as the starting point for the two search algorithms that are developed to improve the solution: repetitive local optimization and tabu search.

The major difference between column generation and the search algorithms is that, in the search algorithms, we replace variables with new ones as we inspect more variables, instead of adding additional variables. This can potentially offer computational time and memory savings.

4.2.1 Greedy Flight Leg Pairing Algorithm

The objective of the D-ARM is to maximize protected revenue. This is achieved when as many as possible of the most valuable itineraries are in the most important layer. If the value of a flight leg is assumed to be proportional to the value of the itineraries that use that flight leg, the routes that are formed in the most important layer will, as a matter of course, have the most valuable flight legs in them. Thus, if we assign equal revenue fractions of an itinerary to its constituent flight legs, and set the value of a given flight leg equal to the maximum value for all the itineraries that use it, we can construct routes of high value by connecting flight legs of similar value.

At first glance, it may appear that we have exchanged one problem – that of exploring all possible combination of routes – for a problem of equal complexity – that of exploring all possible combination of flight legs. However, we can reduce the problem size significantly by exploiting the typical hub-and-spoke structure of major US airlines. Specifically, if we begin by pairing flight legs at spoke airports, where the number of potential connections is considerably less than at hub airports, we significantly reduce the number of flight leg pairings that we need to consider. Thus, although this problem is NP-hard, we find a solution quickly because the problem is very small in size – the spoke airport with the most flight legs has only 54 operations.

In addition, because the connections at spoke airports drive aircraft utilization, holding an aircraft on the ground at a spoke airport will effectively take that aircraft out of use. This occurs because the aircraft will not be in the pool of aircraft at the hub airport it would have flown to, had it not been held at the spoke airport. Thus, at the same time, we can ensure that our solution does not require additional aircraft by constraining the total ground time at each airport to be less than or equal to the total ground time at each airport in the current routing.

The algorithm has four steps, summarized as follows:

STEP 0:	Fix connections for flight legs between spoke airports
STEP 1:	At spoke airports: combine flight legs into flight-leg-pairings, where the constituent flight legs are of similar value
STEP 2:	At hub airports: combine flight-leg-pairings into routes where the constituent flight-leg-pairings are of similar value
STEP 3:	Construct 3-day-long routes

In Step 0, we fix the connections for the flight legs between spoke airports. Analysis of the aircraft routings of a major US airline revealed there were only six such flight legs out of the total of 1,134 flight legs, and that these flight legs were connected to each other in three small cycles. Thus, for the sake of simplicity, these connections were maintained.

In Step 1, we combine the flight legs into flight-leg-pairings – pairs of flight legs from a hub airport to a spoke airport and back to a hub airport, where both hubs may not necessarily be the same. That is, at each spoke airport, we search for the set of connections for which the flight legs have similar value, and the total ground time is less than or equal to the total ground time in the current routing. This problem is modeled as an integer program where i and j are the flight legs arriving at and departing from the airport, respectively. The decision variable w_{ij} has value one when the arriving flight leg i is connected to the departing flight leg j, and zero otherwise. The parameters c_i and c_j are the revenues for flight legs i and j, respectively. The variable t_{ij} is the ground connection time between flight legs i and j. The parameter T is the total ground time at the airport in the current routing. The integer program is formulated as follows:

$$\max \quad \sum_{ij} c_i c_j w_{ij} \tag{4.1}$$

$$\text{s.t.} \quad \sum_{i} w_{ij} = 1 \quad \forall j \tag{4.2}$$

$$\sum_{j} w_{ij} = 1 \quad \forall i \tag{4.3}$$

$$\sum_{ij} t_{ij} w_{ij} \leq T \tag{4.4}$$

$$w_{ij} \in \{0, 1\} \quad \forall ij \tag{4.5}$$

In the objective function (4.1), we multiply the flight leg revenues c_i and c_j, times the value of the decision variable. Thus, the value of the objective function increases significantly when the two highest revenue flight legs are paired together, and so on. The constraints (4.2) and (4.3) are the cover constraints. The constraints (4.2) ensure that for each departing flight, there is exactly one arriving flight to which it is connected. The constraints (4.3) ensure that for each arriving flight, there is exactly one departing flight to which it is connected. The constraints (4.4) ensure that the total ground time is less than or equal to the total ground time in the current routing. The constraint (4.5) ensures that the decision variable is binary.

Theorem 1 *If the total ground time at each airport is not increased, the number of aircraft does not increase as well.*

Proof: Let us show that *if the number of aircraft is increased, the total ground time at that airport is increased.*

Suppose the number of aircraft is increased and the total ground time at that airport is not.

For each airport, we can construct the following graph:

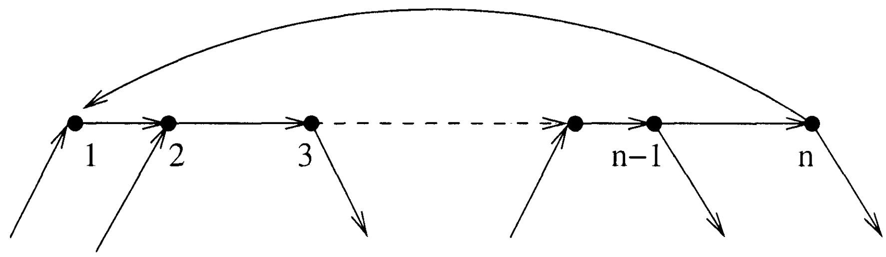

Figure 4-2: Aircraft Flow at an Airport

Let the nodes 1 to n be an event node, which can be either arrival or departure. Let node 1 be a first arrival after the airport is emptied. Let T_i be a time between event i and $i+1$ for $i < n$, and time between event n and 1 for $i = n$. For a given routing, let X_i be the number of aircraft on the ground between event i and $i+1$ for $i < n$, and the number of aircraft on the ground between n and 1 for $i = n$. Let X_n be N. Note that $X_i = X_{i-1} + 1$, if the event i is arrival, and $X_i = X_{i-1} - 1$, if the event i is departure. The total ground time for this airport is

$$\sum_{i=1}^{n} T_i X_i. \tag{4.6}$$

Increasing the number of aircraft by one at any ground arc will increase the number of aircraft by one for all other ground arcs by the balance constraints. Let $\bar{X}$ be the number of aircraft on the ground with an extra aircraft:

$$\sum_{i=1}^{n} T_i \bar{X}_i = \sum_{i=1}^{n} T_i (X_i + 1) = \sum_{i=1}^{n} T_i X_i + \sum_{i=1}^{n} T_i. \tag{4.7}$$

Since $\sum_{i=1}^{n} T_i$ is 24 hours, the total ground time for the new routing is strictly greater than the original total ground time, which contradicts the first statement.

Therefore, if the number of aircraft is increased, the total ground time at that airport is increased. Thus, if the total ground time is not increased at each airport, the number of aircraft in that airport is not increased, either. As a result, the total number of aircraft is not increased.

∎

In Step 2, we combine the flight-leg-pairings into routes. We use the same integer program that is used in Step 1 to build the connections at the spoke airports. The difference between Step 2 and Step 1 is that in Step 2, we use flight-leg-pairings instead of flight legs. Because we already have connections at the spoke airports, we can treat these pairs of flight legs as if they were single flight legs.

In Step 3, we construct routes that are equal to three days in length, to ensure that the aircraft on each route will be able to meet their required maintenance schedule. While this may change the connections at the airport where the longer route is broken, this will not change the total ground time. This approach may overly constrain the maintenance requirement. While it guarantees the maintenance opportunity at least once in every five days, some route may have more maintenance opportunities than required and model does not capture all maintenance feasible routings in the model.

4.2.2 Getting a Initial Feasible Solution

A set of routes which is generated by the greedy flight leg paring algorithm, is fed into D-ARM in Chapter 2. This can be directly solved by CPLEX due to its small number of variables. The route is assigned to a layer by D-ARM. As a result, we assign routes to appropriate layers, thus, all flight legs are assigned in a layer.

Although the greedy flight leg pairing algorithm generates a feasible solution very quickly, it is a myopic rather than a global approach to grouping flight legs together.

That is, because we only consider the possible connections at a given airport, we do not take into account the value of downstream flights. For example, the value of a route may be higher if an arriving flight i is connected to a departing flight j, such that j is not the best connection in terms of the combined value of the single connection in question. The reason for this is that the combined value of flights i, j, and a flight that is downstream j, is higher than any possible combination that could be achieved if i were paired with the best departing flight. In addition, even if we had a solution that globally grouped the flight legs with similar values, this grouping does not necessarily mean the solution had a better D-ARM value, since we assumed that the revenue of an itinerary is proportionally divided among its flight legs.

To compensate for this myopia, we develop two search algorithms to improve the initial feasible solution.

4.2.3 Search Algorithms

Repetitive Local Optimization

The simplest way to achieve the better solution is to swap a sequence of flight legs from a given route with a sequence of flight legs from another route which improves the objective function value. If we keep swapping the improving pairs until there is no more improving one, we get a local optimal solution.

The repetitive local optimization is summarized below.

BEGIN

Select an initial $x \in X$ and let $x^* = x$. Set the iteration counter $k = 0$.

while $k \leq K$ **do**

swapping search algorithm

store solution x_k if $f(x_k) > f(x^*), x^* \leftarrow x_k$

update tabu list algorithm

k = k+1

end while

END

We use the solution from the greedy flight leg pairing algorithm as an initial feasible solution. The routes from the greedy flight leg pairing algorithm are used as the set of routes in D-ARM. For routes we got from the greedy heuristics, the proper layers are assigned by solving D-ARM.

After we have an initial solution, we get a local optimal solution by search algorithms. Once we get a local optimal solution, we update the 'tabu list' so that we get a different local optimal solution in the next iteration. Then we start from the initial feasible solution again and find a local optimal solution by swapping search algorithm. However, since we set our tabu list, the new local optimal solution is not the same as the previous ones. After K iterations, we have K local optimal solutions. All of them start at the same initial solution. However, for each iteration we have a different tabu list, thus a different local optimal solution. Among the local optimal solutions, we can easily pick the best solution by comparing the objective function value.

The swapping search algorithm is summarized below. As the algorithm illustrates, each pair of routes is checked for (1) swapping feasibility, (2) constraint violations, and (3) objective function value improvement. Swap feasibility ensures that a sequence of flight legs from one route can only be swapped with a sequence of flight legs from another route if there are two different instances where the aircraft in both routes are on the ground at the same airport at the same time. If there exists a feasible swapping opportunity, we then check whether, after swapping, both routes will satisfy all the constraints in **D-ARM**, e.g., whether the number of flights in each layer will be less than or equal to the desired number, and whether the number of operations at each hub airport in the first layer will be less than or equal to the capacity at that hub airport in bad weather. If there are no constraint violations, we then check whether the swap will improve the objective function value. For each route, we determine the other route which provides the best improvement in objective function value and then execute the swap for that specific pair of routes. We repeat these until there are no further swapping opportunities that are beneficial.

```
BEGIN
    while Any feasible improving pair exists do
        for each route i do
            for each route j ≠ i do
                Check swapping feasibility
                Check constraint violations
                Check objective function value improvement
            end for
            Swap sequences that improve the objective value the most
        end for
    end while
END
```

At the end of the each swapping search algorithm, the tabu list is updated. The tabu list is a list of swaps that are prohibited. Subsets of the sequences that are swapped during iteration are included in a tabu list. These prohibited swaps are then removed from the tabu list after a specified number of iterations on the list. This is the core of a traditional tabu search algorithm. That is, each time we find an improved solution using the swapping search, we place the solution path to the improved solution on the tabu list for a specified duration, so that as we continue the search we will be forced to search another area for an improved solution.

Tabu Search

A tabu search is often used to solve combinatorial optimization problems because it is adaptive in nature, provides a mechanism to direct searches away from local optima, and is synergistic rather that competitive with other solution methodologies, such as linear programming algorithms or other search algorithms [29] [30]. The idea behind a tabu search is that, by placing the path to a possible solution on a prohibited list, it is possible to escape the trap of local optimality. The tabu search has been used in a wide variety of problems from scheduling to telecommunications to neural networks [31]. Gendreau et al. [28] solved real-time vehicle routing by parallel tabu search. Garcia et al. [27] also used parallel tabu searches to solve vehicle routing problems with time windows. Kelly and Xu [38] solved vehicle routing by a set partitioning based heuristics using tabu searches. In airline scheduling problems, Budenbender et al. [16] combined a tabu search with branch-and-bound to design the direct flight network. Although there are few search algorithms for aircraft routing (for example, Greedy Randomized Adaptive Search Procedure (GRASP) and Direct-search algorithm (DIRECT)), these algorithms are for finding aircraft flight paths [4][11].

One of the strengths of a tabu search is its flexibility; additional constraints can be easily incorporated into its search algorithm. Different algorithms can be easily combined with a tabu search as well. Therefore, by having a tabu search for D-ARM, we keep many possibilities open for constraints we might consider later. There are several more advantages of having a tabu search for D-ARM. First, the number of routes we need to store is the same as the total number of routes created by the greedy flight leg pairing algorithm. Thus, while the flight legs in any specific route may change, we still have the same number of routes, and the same memory requirement. Second, because the swapping search is essentially comparing and swapping sequences between two routes, it is very fast in comparison to an optimization algorithm.

Most tabu searches have a common structure. A tabu search starts with an initial feasible solution, often obtained by a greedy algorithm. Until the stopping condition is satisfied, a tabu search keeps finding the best solution in the neighborhood [32]. A tabu search algorithm for D-ARM is summarized below.

BEGIN

Select an initial $x \in X$ and let $x^* = x$. Set the iteration counter $k = 0$.

while $k \leq K$ **do**

swap the pair which gives the best objective function value

store solution x_k if $f(x_k) > f(x^*), x^* \leftarrow x_k$

update tabu list algorithm

k = k+1

end while

END

A tabu search algorithm is very similar to repetitive local optimization. The major difference is that we move in a direction that gives the best objective function value. That is, we allow the objective function value to be decreased if it is still the best possible one. The pair to be swapped is found by inspecting all potential swaps and picking the best. Note that in the repetitive local optimization, we pick which route to swap for each route, and we pick the best pair among all the possible swaps in the tabu search. Like the repetitive local optimization, possible swaps are identified by checking swapping feasibility and constraint violations. The tabu list is updated in a manner similar to updating the repetitive local optimization. The swapped pair is placed in the tabu list for X iterations. After X iterations, the pair is allowed to be swapped again.

Figure 4-3 shows the trajectory for different X values. The X-axis is the number of iterations, and the Y-axis is the objective function improvement from the initial solution. The trajectory increases fluctuation as the X value increases. As we keep the swap in the tabu search longer, we are more likely to move toward the worse solution in terms of objective function, as that forces the solution to land far from the current best solution. Thus we are more likely to get to the new, better solution. We also found that the objective function value improves slowly as we have more iterations.

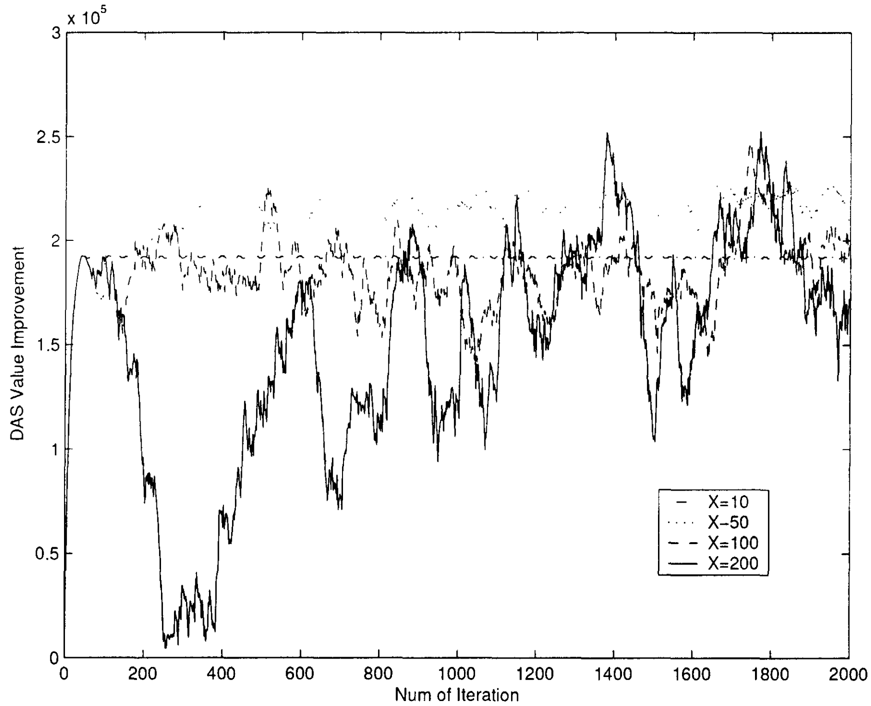

Figure 4-3: Tabu Search

4.2.4 Comparisons between Search Algorithms

Figure 4-4 shows the trajectory for repetitive local optimization and tabu search with $X = 50$ and $X = 100$. The X-axis and Y-axis represent the number of iterations and the objective function improvement from the initial solution, respectively.

As the figure shows, we can observe that for the repetitive local optimization, the solution keeps improving until the local optimal solution is found. Then it starts from zero again, until we find another local optimal solution. We can see that the tabu search generates the better solution in general. This is because we allow searching a neighborhood for a better solution in the tabu search while we stop when we reach the local optimum in the repetitive local optimization. However, note that the scale for DAS value improvement axis is 10^5. The solution from the tabu search is less than 0.5 percent higher than the repetitive local optimization. Also, note that the horizontal axis represents the number of iterations. Although it does not show in the graph, the time it takes per iteration is not the same in both algorithms. In the tabu search, each iteration takes about 0.4 seconds while it only takes 0.05 seconds in the repetitive local optimization. This is because, in the repetitive local optimization, we do not necessarily pick the best pair to swap. Since we select one route and pick the best route to swap with that route, the searching domain is much smaller.

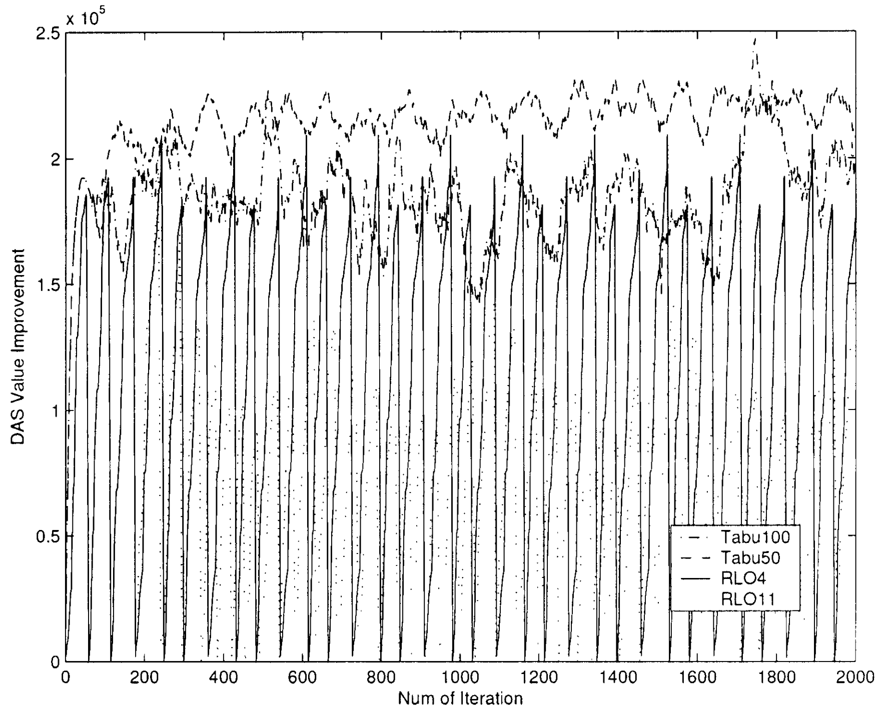

Figure 4-4: RLO vs. Tabu Search

For either search algorithm, the solution is within five percent of an upper bound from D-FAM.

Chapter 5

Model Results

In this chapter, we compare the model results for the three proposed models: degradable schedule partitioning model (D-SPM), degradable fleet assignment model (D-FAM), and degradable aircraft routing model (D-ARM). Using the IP models and solution methodologies discussed in Chapter 2 and Chapter 4, the flight legs are assigned to layers. Below are the results for objective function value, protected revenue, and distribution for itinerary, passengers, and revenue.

Table 5.1 shows the objective function value, and problem size for D-SPM, D-FAM, and D-ARM.

Table 5.1: Model Results Summary

Model	Obj Value	No. Var	No. Const
D-SPM	8,667,632	7,642	7,130
D-FAM	8,508,188	43,030	28,917
D-ARM	7,690,901	many	5,784

The objective function value is higher when we consider a degradable scheduling earlier in the planning procedure. When we consider degradable scheduling earlier in the planning procedure, less constraints such as fleet assignment or routing we have. Thus, the objective function is higher. D-SPM has a manageable number of decision variables and constraints. Thus, solving D-SPM only takes a few seconds for the given problem parameters. In fact, D-FAM has a problem structure similar to D-SPM. D-FAM with two layers and seven fleet types has about the same problem size as D-SPM with 14 layers. Therefore, D-FAM takes more time than D-SPM, which is mainly due to the different problem size. By defining routes as decision variables, we get rid of the balance constraints in D-ARM. However, since the number of variables increases exponentially as the number of flight legs increases, even enumerating all the decision variables is extremely challenging, and that is the main source of its difficulty.

Note that solutions for D-SPM and D-FAM are optimal, whereas D-ARM has the heuristic solution from the tabu search. To analyze the solution quality of D-ARM, we can compare it to an upper bound: the D-FAM solution. Any feasible D-ARM solution satisfies all constraints for D-FAM. However, the other way is not necessarily true because we do not consider maintenance routing in D-FAM. Thus, the objective function value of D-FAM can be used as an upper boundary of D-ARM. Therefore, we can conclude D-ARM is within 10 percent of the optimal solution. Since we chose the solution with a higher FAM cost and a higher degradable airline scheduling value in D-FAM, the upper bound for D-ARM, given the fleet assignment, must be lower than the D-FAM value, which makes the boundaries even tighter.

As stated in the previous section, we reduced the size of the problem by considering only itineraries with more than the average revenue. Thus, to get a complete picture of the improvement that the degradable schedules afford, it is necessary to consider the distribution of itineraries, passengers, and revenue for the entire schedule, as the objective function value is not a complete descriptor of the revenue that is protected, or by extension, the number of passengers that are protected.

Table 5.2 shows the distribution of flight legs. For all models, the number of flight legs in each layer is about the same; 60 percent in Layer 1, 40 percent in Layer 2. This is due to the fact that we have constrained the number of flight legs in each layer.

Table 5.2: Number of Flight Legs

	Layer 1	Layer 2	Total
D-SPM	681 (60.0 %)	453 (40.0 %)	1,134
D-FAM	680 (60.0 %)	454 (40.0 %)	1,134
D-ARM	681 (60.0 %)	453 (40.0 %)	1,134

Table 5.3 and Table 5.4 show the distribution of itineraries and passengers, respectively. Note that we consider the itinerary to be in Layer 1 only when all its flight legs are in Layer 1. This explains why we have only about 49 percent of itineraries in Layer 1 with 60 percent of flight legs. In fact, more than half of the itineraries in Layer 2 have at least on flight leg in Layer 1. However, since they have at least one flight leg in Layer 2, we consider those itineraries in Layer 2. D-SPM, D-FAM, and D-ARM protect 48.7 percent, 49.0 percent, and 48.6 percent of itineraries, respectively. Although one model protects more itineraries than others, the difference is not significant.

Table 5.3: Number of Itineraries

	Layer 1	Layer 2	Total
D-SPM	10,052 (48.7 %)	10,063 (51.3 %)	20,655
D-FAM	10,122 (49.0 %)	10,533 (51.0 %)	20,655
D-ARM	10,043 (48.6 %)	10,612 (51.4 %)	20,655

Passengers are in Layer 1 if their itinerary is in Layer 1. In D-SPM, 67.5 percent of passengers are in Layer 1; in D-FAM 66.3 percent; in D-ARM 62.7 percent. D-SPM protects additional 4.8 percent of passengers with about the same number of itineraries compared to D-ARM. From the table, we can observe that D-SPM protects itineraries with more passengers than itineraries in D-FAM or D-ARM.

Table 5.4: Number of Passengers

	Layer 1	Layer 2	Total
D-SPM	44,982 (67.5 %)	21,661 (32.5 %)	66,643
D-FAM	44,165 (66.3 %)	22,478 (33.7 %)	66,643
D-ARM	41,769 (62.7 %)	24,874 (37.3 %)	66,643

Although we protect about the same number of itineraries, the number of passengers protected by degradable airline schedules is significantly different, even though we do not have the number of passengers explicitly in our optimization models. The reason for this is that there is a positive correlation between the number of passengers and the total revenue for an itinerary. Since an itinerary with more passengers tends to have a higher total revenue, by protecting more revenue, the degradable airline schedules tend to protect more passengers as well.

Table 5.5 shows the distribution of revenue. The revenue in each layer is the total fare of passengers in each layer. D-SPM protects 74.5 percent of the total revenue; 73.3 percent for D-FAM; 66.6 percent for D-ARM. The percentage gap for passengers among the models is greater than the gap for itineraries. This means that D-SPM not only protects the itineraries with more passengers, it indeed protects itineraries with higher revenue.

Table 5.5: Distribution of Revenue

	Layer 1	Layer 2	Total
D-SPM	9,624,460 (74.5 %)	3,287,762 (25.5 %)	12,912,222
D-FAM	9,462,390 (73.3 %)	3,449,839 (26.7 %)	12,912,222
D-ARM	8,600,440 (66.6 %)	4,311,791 (33.4 %)	12,912,222

From Table 5.4 and Table 5.5, we can calculate the average fare for each layer. The total average fare is $193.8. In D-ARM, the average fare for passengers in Layer

1 is \$205.9, while the average fare for passengers in Layer 2 is \$173.3. The gap is even bigger for D-FAM and D-SPM. The average fare for passengers in Layer 1 is \$214.3 and \$214.0, respectively, and the average fare for passengers in Layer 2 is \$153.5 and \$151.8, respectively. As the degradable scheduling value gets higher, the difference between the average fares for layers gets higher, too.

From these results, we can observe that, with the same number of flight legs and similar number of itineraries, the schedule with higher degradable scheduling value protects more passengers, and perhaps equally importantly, protects more revenue. A schedule with a higher degradable scheduling value not only protects more passengers and more revenue, but protects passengers with higher fare. We can conclude that the flight-layer assignment with a higher degradable scheduling value protects the *right flight legs*, given the constraints on the number of flight legs we can protect.

Chapter 6

Simulation Results

To illustrate the benefits that the degradable schedule provides, we used the MIT Extensible Air Network Simulation (MEANS) to simulate the operations of the airline in question on a representative normal weather day and bad weather day to determine the flight and passenger delays and the costs involved.

Table 6.1 shows fleet assignment costs and degradable scheduling values for each model. For the representative airline, there is no feasible fleet assignment, given a flight-layer assignment from D-SPM. Having a layer fixed for each flight leg restricts fleet assignment. However, this does not necessarily mean that D-SPM always does not have feasible fleet assignments and aircraft routing for all airlines. Therefore, in this chapter, only D-FAM and D-ARM are presented as the degradable airline schedules. These two degradable airline schedules are compared to a schedule which is constructed in the traditional way.

Table 6.1: Fleet Assignment Cost and DAS value

	Fleet Assignment Cost	DAS value
D-SPM	8,667,632	infeasible
D-FAM	8,508,188	7,624,983.7
D-ARM	7,665,188	7,612,869.4

There are two ways to construct the *current routing*, or traditional routing. One is from the historical data, which is described in more detail in Appendix A. Since we only took the domestic part of the schedule, constructing routes based on historical data, which includes international flights, is not realistic. The alternative is to assign fleets and construct routes for the given network in the same manner as airline currently do. Thus, a traditional schedule is derived by taking the fleet assignment used to develop D-ARM and then generating aircraft routing using a first-in-first-out criterion when combining flight legs.

6.1 MEANS

MEANS is an event-based simulation of the U.S. National Airspace System (NAS). In MEANS, air traffic flows through a network of 205 major US airports. Each airport in the network is modeled as both a source and sink for air traffic with a capacity profile that captures the tradeoff between arrival and departure rates, as well as the transitions between normal and bad weather. An additional source/sink is used to model all other airports in the NAS. The simulation tracks every flight, approximately 42,000 per simulation day, through the network. MEANS also tracks every passenger individually. Flights are tracked through a series of modules/states, as depicted in Figure 6-1. An aircraft is in the gate state when it is docked at the gate at its origin. When the aircraft pushes back from the gate, it enters the taxi-out state and stays in that state for as long as it takes to taxi to the departure queue at the runway. The aircraft then enters the departure queue state. Upon receiving its departure clearance, the aircraft enters the en-route state. The en-route state encompasses four separate flight phases: take-off, climb-out, cruise and decent. Upon reaching the terminal airspace around its destination, the aircraft enters the arrival queue state which corresponds to both airborne holding before landing and landing itself. The aircraft then enters the taxi-in state from touchdown until the time is reaches the gate at its destination. A second class of modules, such as the air traffic control system command center (ATCSCC) and airline modules, are decision-making

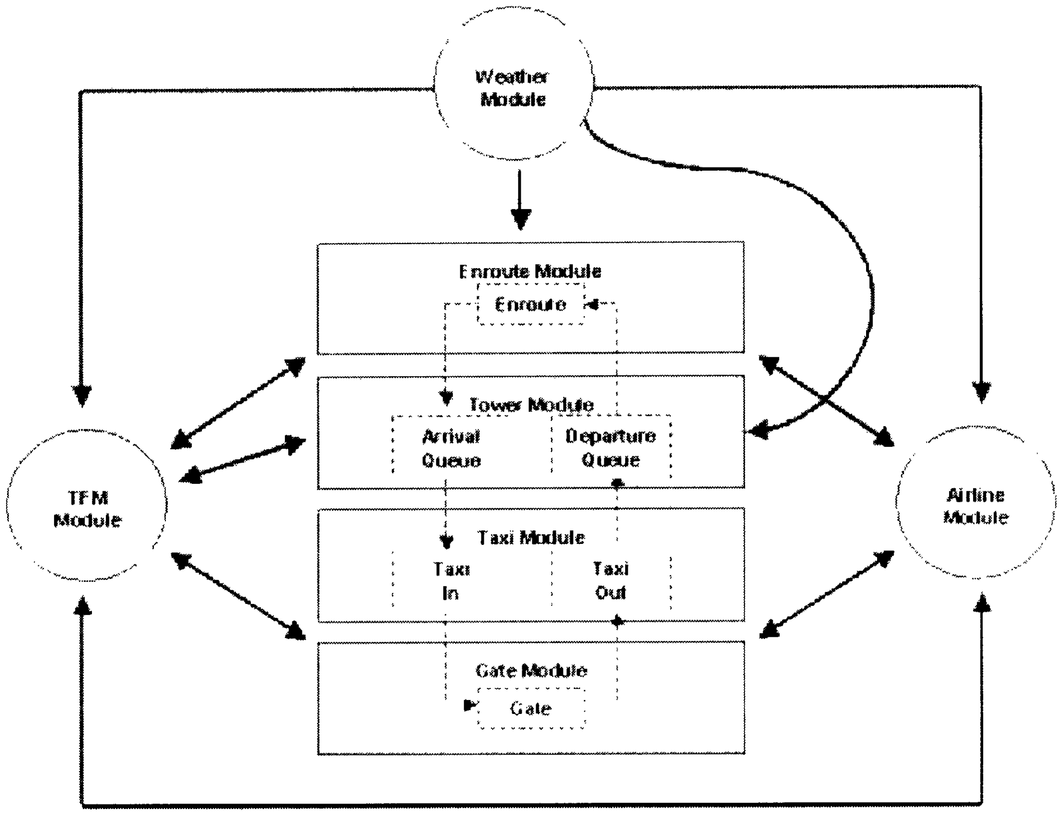

Figure 6-1: Structure of MEANS

modules because they determine the action of the modules that handle flights. A third class of modules, such as the weather module, are information modules because they provide information for other modules to make decisions or change their operating conditions.

The degradable airline schedule requires the real time prioritization of flight legs. The airline agent in MEANS prioritizes flight legs in two ways: by swapping flight leg-to-slot assignments in a ground delay program (GDP) and by swapping positions in the departure queue. These swapping methods ensure that the available capacity, both arrival capacity during a GDP and departure capacity at other times, is preferentially assigned to flight legs in Layer 1. In a GDP, arrival slots are created based on the arrival rate that can be handled by the destination airport. Flight legs are initially assigned to slots based on the planned schedule. Airlines then have the ability to swap their flights within the set of slots which have been assigned to them, moving one flight earlier and moving another later. The airline agent does this swapping automatically, moving the flight legs in Layer 1 forward and thus displacing the flight legs in Layer 2. While a flight leg is not allowed to be moved into a slot earlier than its originally scheduled time, flight legs in Layer 1 are moved forward until there are no feasible slots filled by a flight leg from Layer 2. A similar system is used to swap flight legs in the departure queue to distribute delays caused by limited capacity at the departure airport and to prioritize flights when no GDP is active. Although no formal slots are assigned, MEANS re-orders the flight legs within the departure queue for a given airline based on their priority. Unlike the GDP, there is no constraint on feasiblity of slots based on the original schedule. Thus, every time a new flight is added to the departure queue, the flight legs are reordered so that all of the flight legs in Layer 1 are first. Flight legs are ordered on the departure time of the connecting flights, to other flight legs in the same layer.

In the mode selected for this study, all flight legs in MEANS are allowed up to two hours of delay. If the delay for a flight leg is projected to exceed this two-hour threshold, the flight leg in question is cancelled. When an airline decides to cancel a flight leg, subsequent flight legs in that aircraft's routing are cancelled up to and including the flight leg that would have returned that aircraft to its current location, or until the end of the day. While this is a very simplistic aircraft recovery strategy, it is sufficiently realistic for comparing the performance of the degradable schedule to the traditional schedule. It is assumed that, unless a GDP is active, flight legs will depart on time if the aircraft is available. Passengers who fail to make their connection because their inbound flight leg is delayed beyond a certain threshold are referred to as *disrupted* passengers. Passengers are also considered to be disrupted if the first flight leg in their itinerary arrives on time but the second flight leg is cancelled, or when their first or only flight leg is cancelled. In the simulation, all disrupted passengers are re-accommodated on an alternative itinerary on that day or, if that is not possible, they are added to the group of passengers who are labelled as *not served* at the end of the day. The alternative itinerary is determined using a greedy shortest path heuristic that considers both the seat availability on candidate flight legs at the time of disruption and the shortest path (in terms of time) for the passenger.

Because it is also important to characterize the performance of the NAS within the context of the typical uncertainties observed in the operating environment and in agent behavior, MEANS has a Monte-Carlo mode where a simulation is repeated with different parameter values from user provided distributions. Currently, distributions are allowed for the time an aircraft spends in the taxi-out state, the en-route state, and the taxi-in state, and the delay observed at push back due to internal airline uncertainties unrelated to the delay of the previous flight legs for the aircraft and crew. For each run, the actual value is selected randomly from the user-provided distributions. The results of these simulation runs are then post-processed to provide probabilities and distributions of output parameters, such as delay. The default distributions in MEANS are derived from five years of Airline Service Quality Performance (ASQP) data. The Monte-Carlo mode is selected for this study. Thus, the data presented is the result of 300 MEANS runs of the same day of operations. Three hundred runs was determined in a validation test to be a sufficiently large number of iterations to achieve convergence of all output parameters.

6.2 Flight and Passenger Delay

As previously mentioned in Chapter 1, visual flight rules (VFR) are used in normal weather while instrument flight rules (IFR) are used in bad weather when visibility is poor. Detailed simulation results for all VFR/IFR combinations for three hub airports are presented in Appendix B. In this chapter, we only present the case with 1) VFR at all hub airports (normal weather) and 2) IFR at all hub airports (bad weather) to compare the extreme cases.

Table 6.2 and Table 6.3 show the daily flight statistics for D-FAM, D-ARM, and traditional schedule in the two extreme weather cases. As the tables show, the average flight delay in both weather cases are very similar; about 11 minutes under VFR, about 22 minutes under IFR. While the average delay for D-FAM, D-ARM, and traditional schedule is about the same, the distribution of delay in the degradable schedule is different between layers, the desired effect of having a degradable schedule. As the tables show, under VFR, the delay in Layer 1, the protected layer, is 9 minutes, while the average delay in Layer 2 is 15 minutes. The difference is even greater under IFR; the delay in Layer 1 is 13 minutes, while the average delay in Layer 2 is 35 minutes. It it important to note that in the U.S., a flight must be delayed more than 15 minutes before it is considered *late*. Even though we have large delays for flights in Layer 2, we keep the average delay for flights in Layer 1 under 15 minutes. As a result of differentiating the delay level for flights in each layer, the standard deviation of the flight delay is higher in the degradable airline schedules. In fact, degradable airline scheduling redistributes the delay so that flight legs in Layer 1 are guaranteed less delay and flight legs in Layer 2 have more delay. Thus, overall, the on-time performance is improved in degradable airline schedules because we reduced delay for flights in Layer 1, which is 60 percent of the total flights. Under VFR, the on-time performance for the traditional schedule is 76 percent, while D-FAM has 77 percent and D-ARM has 76 percent. However, under IFR, the difference is significant: 52 percent of the flight legs are on time in the traditional schedule while 57 percent of flight legs are on time in both D-ARM and D-FAM. The redistribution of delays increases the number of flight legs with extremely long delay. As a result, more flight legs are delayed more than two hours, which is the criterion for cancellation. Thus, some cancellations occur in D-FAM and D-ARM under IFR while there are no cancellations in the traditional schedule, as shown in Table 6.3.

While flight delays are the most frequently reported metric of airline reliability, it also important to consider the impact of delays on passengers. Because many passengers have multi-flight leg itineraries, the delay or cancellation of one or more

Table 6.2: Flight Statistics in VFR for All Hub Airports

	D-FAM			D-ARM			Traditional
	Layer 1	Layer 2	Total	Layer 1	Layer 2	Total	Total
No. of Flights	680	454	1134	681	453	1134	1134
No. of Flights Cancelled	0	0	0	0	0	0	0
Avg Flight Delay (min)	9	15	12	9	15	11	11
StdDev Flight Delay (min)	18	31	24	16	28	22	21
Pr (on-time)	0.80	0.72	0.77	0.80	0.71	0.76	0.76

Table 6.3: Flight Statistics in IFR for All Hub Airports

	D-FAM			D-ARM			Traditional
	Layer 1	Layer 2	Total	Layer 1	Layer 2	Total	Total
No. of Flights	680	454	1134	681	453	1134	1134
No. of Flights Cancelled	0	13	13	0	8	8	0
Avg Flight Delay (min)	13	36	22	13	35	21	22
StdDev Flight Delay (min)	20	36	30	19	35	29	26
Pr (on-time)	0.70	0.38	0.57	0.69	0.40	0.57	0.52

flight legs in their itinerary can result in missed connections, long wait times at either originating airport or an intermediate airport, and disproportionate delays. The daily passenger-itinerary information is reconstructed based on the average passenger-itinerary information from the representative airline. Since we used the average daily data from a month, some itineraries has the fractional number of passengers. When an itinerary have a fractional number of passengers, we simply round the number to the closest integer value. As a result, the number of total passengers in the simulation does not match exactly the number of passengers in the optimization.

Table 6.4 and Table 6.5 shows the daily passenger statistics for D-FAM, D-ARM, and the traditional schedule in the two extreme weather cases: VFR in all hub airport and IFR in all hub airports. The passenger delay statistics are similar to that of flights, but benefit of Layer 1 is more significant. By definition, passengers in Layer 1 have all flight legs in their itineraries in Layer 1. With all flights with small delays, they are likely to have less overall delay. However, passengers in Layer 2 have various cases. If passengers in Layer 2 have all their flight legs in Layer 2, since every flight leg is delayed, passengers may not miss the connections. However, in that case, there will be a considerable arrival delay. If the first flight leg is in Layer 1 and the second leg in Layer 2, even though a passenger is likely to get to a hub airport on time, and make the connection for the next flight leg; arrival delay is still expected because the second leg is in Layer 2. If a passenger has a first leg in Layer 2 and second leg in Layer 1, since the first leg is likely to be delayed and the second leg is likely to depart on time, a passenger has a higher chance of missing the connection than in the traditional schedule.

We can observe a higher passenger delay in Layer 2 than delay in Layer 1. Under VFR, passengers in Layer 1 can expect a 9 minute of delay while passengers in Layer 2 can expect an 18 minute of delay. The total average passenger delay is similar across the schedule, 12 minutes. The difference between the average passenger delay in bad weather is even more significant. Passengers in Layer 1 only expect a 13 minute of delay, similar to the total average passenger delay in normal weather, while passengers

Table 6.4: Passenger Statistics in VFR for All Hub Airports

	D-FAM			D-ARM			Traditional
	Layer 1	Layer 2	Total	Layer 1	Layer 2	Total	Total
No. of Total Pax	43,349	21,674	65,023	40,955	24,068	65,023	65,023
No. of Pax miss connection	511	952	1463	431	1015	1446	1520
No. of Pax didn't get served	143	230	373	124	201	325	343
Avg Pax Delay (min)	9.3	17.4	12.0	8.6	18.4	12.2	12.4
Pr (on-time)	0.81	0.74	0.79	0.82	0.72	0.78	0.77

Table 6.5: Passenger Statistics in IFR for All Hub Airports

	D-FAM			D-ARM			Traditional
	Layer 1	Layer 2	Total	Layer 1	Layer 2	Total	Total
No. of Total Pax	43,349	21,674	65,023	40,955	24,068	65,023	65,023
No. of Pax miss connection	790	2889	3679	714	3051	3765	3112
No. of Pax didn't get served	174	505	679	157	501	658	569
Avg Pax Delay (min)	13.5	45.4	24.1	13.0	45.4	25.0	23.5
Pr (on-time)	0.72	0.43	0.62	0.72	0.43	0.62	0.57

in Layer 2 can expect a 45 minute of delay. The average delay is about 24 minutes.

We can see differences in the service level, i.e. the number of passengers that miss their connections, the number of passengers that are reaccommodated between Layer 1 and Layer 2. But again, the difference is more remarkable in bad weather. The probability that a random passenger misses a connection is about five percent in the traditional schedule and about 5.7 percent in the degradable schedules. However, if the passenger is in Layer 1, the probability is less than two percent. About one percent of passengers are not able to get reaccommodated, thus are not able to fly that day. However, if the passenger is in Layer 1, the probability is less than 0.4 percent, which is not much worse than that in normal weather, 0.3 percent.

Under IFR, we can observe there are more passengers with missed connections, and more did not get served in the degradable airline schedules than the traditional schedule. That is due to more cancellations in the degradable airline schedules. Thus, we lost more revenue in degradable schedules.

However, since cost, such as fuel and maintenance, is reduced by not flying, it is reasonable to consider cost saving as well as revenue lost. There are three cost factors we consider: revenue lost, operating cost saving, and delay crew costs. The revenue lost is simply the summation of fares of passengers who did not get served until the end of the day. When a flight is cancelled, it reduces the cost of fuel and maintenance. We assume crews are still paid whether or not the flight was flown. Fuel cost and maintenance cost for different fleet type is derived from Form 41. When crews are delayed, there are extra labor expenses. For cockpit crews, the different cost are applied for different fleet type. We assume the labor cost for flight attendances are the same no matter which fleet type it is. The total cost is calculated by revenue lost minus operating cost save plus delay crew cost.

Table 6.6 and Table 6.7 show the total cost for D-FAM, D-ARM, and the traditional schedule in two extreme weather cases. Under IFR, the revenue lost is much higher in the degradable schedules than that in the traditional schedule since we cancel flights in the degradable schedules and do not in the traditional schedule. Under VFR, since there is no flight cancelled, there are no savings in aircraft operating cost, while there is remarkable savings from cancellations under IFR. In either case, we can observe slightly higher delay crew costs, in the degradable airline schedules than the traditional schedule. Under VFR, with higher delay crew costs and more passengers who do not get served, the total cost is higher in D-FAM.

The total cost for D-ARM and the traditional schedule is about the same. However, the average fare for passengers who do not get served is about three percent lower in D-FAM compared to D-ARM and the traditional schedule, which have about the same average fare for passengers who do not get served. Under IFR, we have higher revenue lost and higher delay crew costs for the degradable airline schedules. But with the aircraft operating cost savings, the total cost is lower in the degradable airline schedules. In addition, as under VFR, the average fare for passengers who do not get served is lowest in D-FAM. The traditional schedule has the highest average fare for passengers who do not get served.

Table 6.6: Total Cost in VFR for All Hub Airports

	D-FAM	D-ARM	Traditional
Revenue lost	75,503.8	67,527.8	71,236.5
Operating Cost Saving	0	0	0
Delay Crew Cost	225,698	222,081	218,747
Total Cost	301,201.8	289,608.8	289,983.5

As these results show, we successfully differentiate service levels between Layer 1 and Layer 2 for both flight and passengers. When the weather is normal, there are still differences in service levels between Layer 1 and Layer 2, but this does not necessarily save money for a airline. However, when the weather gets bad, the difference is more remarkable. The flight delay is slightly increased in the degradable airline schedules, resulting in higher delay crew costs. Also, since by redistributing

Table 6.7: Total Cost in IFR for All Hub Airports

	D-FAM	D-ARM	Traditional
Revenue lost	126,661	124,750	108,816
Operating Cost Saving	42,129.2	33,308.5	0
Delay Crew Cost	426,417	418,308	422,322
Total Cost	510,948.8	509,749.5	531,138

delays, we have some flights with very high delay, forcing them to be cancelled in the degradable airline schedules. With more cancellations, we serve fewer passengers than the traditional schedule, which does not have any cancellation; therefore more revenue is lost. However, the aircraft operating cost saving from cancellations make up for other costs. Thus, overall, the degradable airline schedule has lower recovery cost.

There is another important point to make about the average fare for passengers who do not get served for each schedule. The objective function for the degradable airline schedule is to maximize the protected revenue, in our particular example, to maximize the revenue in Layer 1. The result shows that, despite a higher number of passengers who do not get served, and higher revenue lost, the average fare for passengers who cannot fly is lowest in D-FAM, and highest in the traditional schedule. This shows that passengers or itineraries with more revenue are better protected as the degradable airline schedule value gets higher. The service quality for protected passengers is also high no matter how the weather changes.

Chapter 7

Sensitivity Analysis

In this chapter, we discuss what would be altered if we changed some of our input parameters. In Section 7.1, we present the trade off between solution time and solution quality. In Section 7.2, we discuss what would happen if we increased the number of layers. In section 7.3, we analyze how we can incorporate factors other than revenue, such as market protection and market share, in the model. In Section 7.4, we discuss what kind of network can benefit more from the degradable airline scheduling.

7.1 Number of Itineraries

We considered only 1,744 itineraries out of 20,655 itineraries in the models. Even though we only consider 8 percent of the itineraries, 80 percent of the passengers and 84 percent of the revenue are captured. The assumption is that because we captured a significant fraction of the revenue with a small number of itineraries, the solution will not be too far from the optimal solution with all itineraries, and there will be a shorter solution time.

To prove our assumption, we solved D-SPM with a different number of itineraries. The reason we only picked D-SPM is because D-SPM is solved in the shortest time, compared to D-FAM or D-ARM, because of its simplicity. Since D-FAM and D-ARM has similar properties but with more constraints and decision variables, we anticipate that the results for D-SPM accurately represent the trade-off between the solution time and the solution quality for degradable airline schedules.

Table 7.1: Sensitivity Analysis for Number of Itineraries in the Model

No. Itn	Prot. Rev	Gap(%)	No. Const	No. Var	No. Itr	Sol. Time(sec)
1,744	9,624,460	0.86	7,130	7,642	1,140	4.51
5,000	9,697,140	0.12	12,818	10,898	4,716	2.65
10,000	9,707,600	0.01	20,343	15,898	18,851	242.07
15,000	9,707,760	0.01	27,254	20,898	92,252	1,067.24
20,655	9,708,520	0	34267	26,553	359,898	2,651.80

Table 7.1 shows the quality of solutions and the solution time for a different number of itineraries. The gap represents the difference between the protected revenue with a given number of itineraries and the optimal solution with all itineraries in the model. With 1,744 itineraries, the solution within 0.86 percent is obtained in a few seconds. With 5,000 itineraries, we get the solution within 0.12 percent in a few seconds as well. The shorter solution time for the model with 5,000 itineraries than the solution time for that with 1,744 itineraries are due to the fluctuation of computational time. However, as the number of itineraries gets higher than 10,000, the solution time explodes, while the gap is about 0.01 percent. If we used all itineraries, the solution time for D-SPM is 2,651 seconds; 44 minutes might be acceptable, given that we do not need to solve D-SPM in real-time or often. However, as the model gets more complicated, such as D-ARM or D-FAM, the solution time may not be reasonable or we may face a memory problem as well.

The marginal improvement for adding more itineraries gets smaller as we add more itineraries. Note that we rank all itineraries and add high revenue itineraries first. After adding a number of itineraries, those left have very low revenue. Therefore, the marginal increase of revenue included in the model gets smaller. We include 84.0 percent of revenue in the model with 1,744 itineraries; 92.2 percent with 5,000 itineraries; 96.9 percent with 10,000 itineraries; 99.0 percent with 15,000 itineraries.

In fact, the itineraries remaining after adding 5,000 itineraries have a revenue of \$64 per itinerary on the average. As the number of itineraries in the model gets higher, the problem size is proportionally increased. However, since the revenue in the model barely increases compared to the increase in the number of itineraries, the solution quality is not improved significantly while the solution time increases dramatically.

One may argue using that 5,000 itineraries instead of 1,744 itineraries may be preferable in terms of solution quality. But note that D-SPM is the simplest model among all degradable airline scheduling models. As it has more decision variables to begin with in D-FAM, as Table 7.1 shows, the break point may be sooner than in D-SPM.

7.2 Layers

In this thesis, we only have two layers. It is definitely possible to have more layers as the number of layers is an input parameter. By having multiple layers, we can differentiate layers within the protected layer as well. If we have multiple layers in protected layers, they will not be cancelled. However, among those flight legs, we can still prioritize flight legs. Suppose an airport was closed for few hours. There is no way to operate any flights in the airport during the closed hours. However, as soon as the airport starts operating, the flight legs in the highest priority layer will have priority. Even during normal weather, flight legs in higher priority layers have priority over flight legs in lower priority layers if they are in the protected layer or not.

With multiple layers, we can observe a similar result. The flight legs in the highest layer will expect the shortest delay and those in the lowest layer will expect the longest delay. This is the same for passengers. Therefore, the standard deviation for the delay may increase even further.

As the number of layers increases, the number of decision variables also increases linearly. Therefore, for D-SPM and D-FAM where we directly feed these models into CPLEX, having more layers has a direct impact on the solution time. For a simple model like D-SPM, we may be able to achive an optimal solution. However, for D-FAM which has more decision variables and constraints, we may need to accept near optimal solution or develop a heuristic one. More layers do not change the solution time for D-ARM as much as in D-SPM or D-FAM. It may take longer to get an initial feasible solution. However, no matter how many layers we have, the number of pairs we need to inspect in search algorithms is in order of R^2, where R is the number of routes. Since the number of routes remains the same, although the solution time may increase, it is not nearly as significant as in other models.

Since we are considering only two layers in this thesis, one layer is automatically a protected layer and the other is not. That leaves us no choice about the size of the layers because the size of protected layer is restricted by airport capacity. However, if we consider multiple layers, we must determine the size of each layer. Like the number of layers, the size of each layer is also an input parameter. We need to make sure to keep the size of each layer big enough to have feasible aircraft routing. In an extreme example, if we have 400 layers of equal size, each layer will have 3-4 flight legs. In that case, we do not have that much flexibility to route aircraft. Since we do not allow inter-layer aircraft connections, this may require additional aircraft or crews.

7.3 Cost Coefficient

In our model, the cost coefficient for an itinerary in Layer 1 is simply the revenue, while it is zero if in Layer 2. There are a few variations we can consider. For the protected market, an airline cannot cancel flights. Usually, the flights in those market are government subsidized, and are very likely to have a lower revenue. Therefore, if we strictly use the revenue as our criterion, the flight legs in the protected market is likely to be in Layer 2 and can be cancelled if necessary. To avoid the cancellation, we can simply put flight legs in a protected market in one of the protected layers. In this way, we can guarantee that the market is served.

There is also a question of market share. For example, if both itinerary a and itinerary b have the same revenue, it is not necessarily true that they are equally important or valuable. If itinerary a is one of 30 other itineraries for the O-D market, and itinerary b is the only itinerary serving the O-D market, we might want to give a weight to itinerary b because passengers in itinerary a are more likely to be reaccommodated if they miss the connection or the itinerary is cancelled.

It is safe to assume that all revenue is protected in the highest priority layer, but, being in lower or even the lowest priority layers does not mean that no revenue is protected. In our model, we assume Layer 1 protects all revenue and Layer 2 protects none. If we have multiple layers, we need to decide how much revenue is protected for each layer. This could be due to on-time operation, or we can simply use layer size as well. As long as it is continuously decreasing as the priority gets lower, we can apply the solution approaches in Chapter 4.

In this thesis, we maximized the total protected revenue, assigning flight legs such that itineraries with higher revenue are in protected layers, and vice versa. The results show that we successfully differentiated services and itineraries. However, when we take cost into consideration, degradable schedules are not always better. We can take other factors, such as delay or cost, into consideration for the coefficient as well.

7.4 Network Structure

Assuming an airline has chosen the airports it wants to serve, there are a number of ways to construct the network. One extreme is a complete network where each airport has direct flights to and from every other airport. The other extreme is a hub-and-spoke network with one hub airport, where all flights are connected only through the hub. The complete network approach requires too many flights. Given limited demand between two origin-destination markets, it is impossible to operate point-to-point service for all markets. Even though the one hub model minimizes total operating costs, the network is often not reliable, as there are no other way to serve the market if the hub airport fails to operate properly. Therefore, most airlines have a network that combines a hub-and-spoke network that includes several hub airports with some point-to-point flights.

An airline with more concentrated hub-and-spoke networks can benefit more from the degradable airline scheduling for two reasons. The first reason is that, if an airline has more concentrated hub airports, there are many aircraft and crew connections made at the hub. If there are aircraft and crew connections made at the same time, it is more likely to have delay propagation. Another reason is that, if an airline has more flights at hub airports, there is more flexibility to differentiate the service. If there is only one flight a day, there is little we could do to degrade the system. However, if there are a few hundred flights a day at one airport, we have the flexibility to build a degradable system that protects important flights and sacrifices less important flights if necessary.

Chapter 8

Conclusion

In this thesis, we define the degradable airline schedule as a schedule that is divided into several independent sub-schedules, or layers, with each layer at a different level of importance. To implement this concept, we identify different stages which incorporate degradable airline scheduling into the current airline schedule procedure. This procedure includes 1) the degradable schedule partitioning model (D-SPM), incorporating degradability *before the fleet assignment*, 2) the degradable fleet assignment model (D-FAM), incorporating degradability *with the fleet assignment*, and 3) the degradable aircraft routing model (D-ARM), incorporating degradability *with the aircraft routing*.

The D-SPM, which has a relatively simple problem structure and small problem size, is fed directly into CPLEX. In the D-FAM, we consider two optimizations: minimizing the fleet assignment cost and maximizing the total protected revenue. We maximize the total protected revenue for which a given fleet assignment cost represents an upper bound. After solving the problem with different given fleet assignment costs, we find a trade-off curve between the fleet assignment cost and the protected revenue. We pick one solution that gives a small increase in fleet assignment cost and a big increase in the protected revenue. In the D-ARM, we define decision variables as routes to incorporate maintenance feasibility into the variables. Since enumerating all possible decision variables is extremely challenging, we develop two search algorithms: repetitive local optimization and tabu search. In both search algorithms, we start with an initial feasible solution from a greedy-type heuristic; in the repetitive local optimization, we start at the same point and gradually move toward a solution with a better objective function value. We then find many local optimal solutions and pick the best one. In the tabu search, we move to a solution with the best objective function value, which may or may not improve the objective function value to find a better solution in the neighborhood.

We found that the solution from the tabu search is 0.5 % better than one from the repetitive local optimization. However, the repetitive local optimization was about ten times faster than the tabu search. The results show that for all degradable airline schedules, there is a clear distinction between Layer 1, the protected layer, and Layer 2, the unprotected layer. It also shows that a schedule with a higher degradable scheduling value protects more itineraries with a higher revenue. The question remains, which of the models would be best to implement?

If we only consider the degradable scheduling value, D-SPM would be our answer, since this approach would be the earliest opportunity to incorporate the degradable airline scheduling into the airline schedule procedure, therefore it is the least constrained. However, it is not that simple. D-SPM does not necessarily guarantee a feasible fleet assignment. It is even possible for that fleet assignment to carry an unreasonable cost. D-FAM has a higher degradable scheduling value than D-ARM, but there is a trade-off. Since we do not consider the degradable airline scheduling when we solve FAM for D-ARM, we have a fleet assignment that minimizes the fleet assignment cost in D-ARM. It is possible to make up the extra cost from D-FAM by revenue management, including pricing. However, for the first step, D-ARM would be a better approach. Since D-ARM is solved at the far end of the planning procedure, another advantage of using D-ARM, especially if the criterion is revenue, is that we can better predict the revenue for itineraries.

Although the degradable airline scheduling is developed as a planning tool, an airline can also use it as an operational tool. Instead of making the layer information public, we can use it as an internal information for a delay/cancellation policy. In that case, we have actual data for revenue on the day of operation; therefore, there is a lower cost for irregular operations in bad weather. However, the problem with this approach is that it provides a lower service quality to some passengers without their knowlege.

In summary, the contributions of this thesis are two-fold:

1) We introduced and implemented the concept of degradable airline scheduling, a schedule with several independent sub-systems with different priorities, and different performance levels. We successfully divide the schedule into several independent sub-schedules, with each sub-schedule having a different flight and passenger performance level.

2) We applied search algorithms to solve aircraft routing problems. In the aircraft routing models, routes are often used as the decision variables to pass some of problem complexity to its sub-problem. This results in the large number of decision variables. One of biggest differences of a tabu search, compared to a column generation, is that in the tabu search, we do not increase the number of decision variables over time in the model. We inspect potential variables and include some of them in the model – but only by replacing them with variables currently in the model. Another difference is that the core operation in the tabu search is comparing and swapping, which is much simpler than optimization.

Thus, we have proved the concept of the degradable airline scheduling and applied the tabu search to the aircraft routing problem.

Looking ahead, it is clear that research in this field, which is still in its infancy, can be developed in many directions. One such direction is a consideration of crews. If the crew scheduling problem is solved after the degradable airline scheduling, then we may have higher crew costs. If the degradable scheduling is solved after the crew scheduling, then we may have a lower total protected revenue. However, even the lower total protected revenue may actually protect more revenue compared to not having a degradable airline schedule. Secondly, we can develop a pricing scheme for different layers. As previously shown, we successfully differentiate the service quality, and therefore we can segment the market based on passengers' preference for reliability. In the first direction, it is likely that the cost will increase; in the second, the profit will definitely increase.

Finally, the concept of the degradable system could be applied to the other areas. The degradable system not only guarantees the system robustness but also differentiates the service levels. Especially for a system directly connected to end users, this degradable system could very likely improve satisfaction for both users and system managers.

Appendix A

IP Model for Constructing Planned Routing

One very convincing way to show the operational benefits of new aircraft routing is to simulate operations using both the proposed aircraft routing and the current planned aircraft routing, and then compare their operational performances. In many instances, however, information about airlines' planned aircraft routes is not available because no such a public database exists, and airlines are reluctant to give out this information. Researchers often get a routing from an observation of the actual airline operations. By simply tracking aircraft tail numbers, we can get a routing. However, because of uncertainties in airline operations, the sequence we get from observed operations is not necessarily the same as what is planned, because of uncertainties, such as the weather or mechanical problems.

Airline operations are recorded in Airline Service Quality Performance (ASQP). ASQP is a database of flight operations for U.S. major airlines. ASQP includes origin, destination, scheduled departure/arrival time, actual departure/arrival time and a tail number for each flight for each day. By connecting fights with the same tail number in ASQP, we can reconstruct sequences of flights. However, as mentioned earlier, airlines change aircraft routings during the operation due to the recovery procedure from delays and cancellations. Thus, the sequence we found from ASQP tail numbers is what actually happened, but not necessarily what was planned. Sometimes this routing may not even be a feasible routing. For example, a flight with an actual arrival time at 6pm can be connected to a flight with an actual departure time is of 7pm. However, if the planned arrival for the former flight is 6:20pm and the planned departure time for the later flight is 6:10pm, this connection will not be the part of the planned routing.

Aircraft routing is, in other words, a collection of pair-wise connections. If we know what the connecting flight for each fight will be, we can construct a routing by putting all pair-wise connections together. Thus, in our model, we would like to find a connecting flight for each flight. This procedure will eventually construct a routing and make a model tractable.

In our model, we have made three assumptions about airline operations; (1) an airline has a daily schedule that is repeated every day, (2) an airline operates as close to the planned routes as possible, and (3) there is only one fleet type. The first assumption is commonly used in airline operations, although a weekend schedule is slightly different from a weekday schedule in a real world. The second assumption means that if we observe the actual airline operation over time, planned connections are likely to be occurred more than those not planned. The third assumption is practical because if we have multiple fleet types. we can simply have a separate network for each fleet type and find routings on each network.

We collect the actual pair-wise connection data in multiple days and find a routing that matches the actual operation as much as possible by an integer programming model. The actual connection data can be found in ASQP. From the data, we can find how many times each pair-wise connection occurred. This information shows some connection patterns. For example, some flights are connected to certain other flights all the time, whereas some connections are made only once or twice during the period. If a connection occurs all the time during the sample period, this means the connection is very likely to have been planned. However, if a connection is made only one time during the period, this connection is very likely to have happened resulted from the recovery process, not have been planned. Suppose there are two incoming flights, A and B, and two outgoing flights, C and D at an airport, and we observe the connections at the airport for 10 days. Suppose flight A is connected to flight C in 9 days out of 10 days, flight B is connected to flight D in 7 days, flight A is connected to flight D in one day, and flight B is connected to flight C in two days. In this example, we can conclude that the connection from A to C is very likely to be planned while the connection from A to D is not. Given a feasible routing, we can calculate the frequency of each connection in the routing. We can compare routings in terms of the number of actual connections captured by each routings. A routing with more overlap is more likely to be a planned one. With the same sample network mentioned earlier, we only have two possible connection plans for this airport. One is to connect A to C and B to D, and the other is to connect A to D and B to C. In the former case, we capture 16 connections out of 19 connections that occurred in 10 days. In the latter case, we capture only three. Thus we can conclude the connecting A to C, and B o D is the aircraft connection plan that was used.

We construct an IP model to find a feasible routing that matches the actual operation as much as possible. The formulation is the following:

$$\max \quad \sum_{i,j} c_{ij} x_{ij}^{k} \tag{A.1}$$

$$\text{s.t.} \quad \sum_{j} x_{ij} = 1 \quad \forall i \tag{A.2}$$

$$\sum_{j} x_{ji} = 1 \quad \forall i \tag{A.3}$$

$$x_{ij} \in \{0,1\} \quad \forall ij \tag{A.4}$$

Let i be incoming flight and j be outgoing flight. Since we are to find pair-wise connections only, our decision variables are defined for each feasible pair-wise connection. The decision variable x_{ij} is binary which has value one if incoming flight i is connected to outgoing flight j, and zero otherwise. Let c_{ij} be the number of times the connection i to j is actually made in the historical data. The objective function (A.1) is the number of actual connections captured by the set of pair-wise connections. The constraints (A.2) and (A.3) ensure the feasibility of connections, and are basically a assignment problem. By maximizing the objective function value, we can have a feasible routing that captures the current operations most, i.e., planned routing.

Appendix B

Simulation Results

In this chapter, detailed simulation results are presented. We run simulations for eight different weather cases. As previously mentioned, the representative airline has three hub airports. Each airport can be either under VFR (normal weather) or IFR (bad weather). Two extreme weather cases – all hub airports with VFR and all hub airports with IFR – are presented in Chapter 6. The flight statistics, passenger statistics and cost for the rest of the six weather cases are presented below.

From the Tables, we can see the clear distinction between Layer 1 and Layer 2 in terms of service level in both flights and passengers. In all weather scenarios, the average delay for flights and passengers in Layer 1 is lower than the average delay for flights and passengers in Layer 2. The difference gets even bigger when weather goes worse. In terms of cost saving, the degradable airline schedules do not necessarily guarantee lower cost when we consider revenue lost, cost of delay, and aircraft operation cost saving because of cancellations. The degradable airline schedules tend to save money when the hub airport with high capacity reduction is affected by bad weather. However, the important point is that in all weather scenarios, the on-time performance and the average delay for passengers in Layer 1 are not only lower than the average but do not depend too much on weather. We also found that the average fare for passengers who did not get served until the end of the day has a negative correlation with the degradable schedule values, or the total protected revenue.

Table B.1: Flight Statistics when HUB1 u VFR and HUB2, HUB3 under IFR

	D-FAM			D-ARM			Traditional
	Layer1	Layer2	Total	Layer1	Layer2	Total	Total
No. of Flights	680	454	1134	681	453	1134	1134
No. of Flights Cancelled	0	13	13	0	6.1	6.1	0
Avg Flight Delay (min)	12	22	16	11	25	17	16
StdDev Flight Delay (min)	20	33	26	18	35	27	24
Pr (on-time)	0.73	0.62	0.69	0.73	0.59	0.68	0.65

Table B.2: Passenger Statistics when HUB1 under VFR and HUB2, HUB3 under IFR

	D-FAM			D-ARM			Traditional
	Layer1	Layer2	Total	Layer1	Layer2	Total	Total
No. of Total Pax	43,349	21,674	65,023	40,955	24,068	65,023	65,023
No. of Pax miss connection	557	1,479	2,037	444	1,462	1,905	1,635
No. of Pax didn't get served	141	234	375	116	204	320	316
Avg Pax Delay (min)	12.3	28.4	17.7	11.3	30.2	18.3	17.3
Pr (on-time)	0.74	0.65	0.71	0.76	0.61	0.70	0.67
Revenue Lost	30,701.1	43,676.4	74,377.5	26,082.6	38,986	65,068.7	65,160.1
Operating Cost Saving	N/A	N/A	42,129.2	N/A	N/A	28,987.9	0
Delay Crew Cost	N/A	N/A	306,927	N/A	N/A	325,765	312.235
Total Cost Lost	N/A	N/A	339,175.3	N/A	N/A	361,845.8	377,395.1

Table B.3: Flight Statistics when HUB2 under VFR and HUB1, HUB3 under IFR

	D-FAM			D-ARM			Traditional
	Layer1	Layer2	Total	Layer1	Layer2	Total	Total
No. of Flights	680	454	1134	681	453	1134	1134
No. of Flights Cancelled	0	10	10	0	3	3	0
Avg Flight Delay (min)	12	29	19	12	28	18	18
StdDev Flight Delay (min)	20	34	28	19	32	26	25
Pr (on-time)	0.74	0.47	0.63	0.73	0.47	0.62	0.60

Table B.4: Passenger Statistics when HUB2 under VFR and HUB1, HUB3 under IFR

	D-FAM			D-ARM			Traditional
	Layer1	Layer2	Total	Layer1	Layer2	Total	Total
No. of Total Pax	43,349	21,674	65,023	40,955	24,068	65,023	65.023
No. of Pax miss connection	778	2747	3525	706	2792	3499	3067
No. of Pax didn't get served	176	511	687	164	534	698	596
Avg Pax Delay (min)	11.8	39.4	21.0	11.4	36.0	20.5	19.9
Pr (on-time)	0.76	0.50	0.67	0.76	0.51	0.67	0.65
Revenue Lost	36,641.1	92,306.2	128,947	34,771.2	97,713.4	132,485	114,789
Operating Cost Saving	N/A	N/A	30,392	N/A	N/A	21,510.6	0
Delay Crew Cost	N/A	N/A	360,975	N/A	N/A	350,735	353.888
Total Cost Lost	N/A	N/A	459,530	N/A	N/A	461,709.4	468.677

Table B.5: Flight Statistics when HUB3 under VFR and HUB1, HUB2 under IFR

	D-FAM			D-ARM			Traditional
	Layer1	Layer2	Total	Layer1	Layer2	Total	Total
No. of Flights	680	454	1134	681	453	1134	1134
No. of Flights Cancelled	0	3	3	0	3	3	0
Avg Flight Delay (min)	12	34	21	12	34	21	20
StdDev Flight Delay (min)	19	37	30	18	38	30	25
Pr (on-time)	0.72	0.39	0.59	0.71	0.41	0.59	0.55

Table B.6: Passenger Statistics when HUB3 under VFR and HUB1, HUB2 under IFR

	D-FAM			D-ARM			Traditional
	Layer1	Layer2	Total	Layer1	Layer2	Total	Total
No. of Total Pax	43,349	21,674	65,023	40,955	24,068	65,023	65,023
No. of Pax miss connection	676	2,496	3,232	688	2,784	3,472	3,018
No. of Pax didn't get served	117	501	679	158	486	644	572
Avg Pax Delay (min)	12.1	39.4	21.2	11.7	41.4	22.7	21.4
Pr (on-time)	0.74	0.46	0.65	0.74	0.46	0.64	0.60
Revenue Lost	37,226.6	90,388	127,615	33,927.3	89,493.7	123,421	109,694
Operating Cost Saving	N/A	N/A	11,737.2	N/A	N/A	7203.0	0
Delay Crew Cost	N/A	N/A	405,892	N/A	N/A	407,146	382,563
Total Cost Lost	N/A	N/A	521,769.8	N/A	N/A	523,364	492,257

Table B.7: Flight Statistics when HUB1 under IFR and HUB2, HUB3 under VFR

	D-FAM			D-ARM			Traditional
	Layer1	Layer2	Total	Layer1	Layer2	Total	Total
No. of Flights	680	454	1134	681	453	1134	1134
No. of Flights Cancelled	0	0	0	0	0	0	0
Avg Flight Delay (min)	11	28	17	10	26	17	16
StdDev Flight Delay (min)	19	35	28	18	33	26	24
Pr (on-time)	0.76	0.48	0.65	0.75	0.50	0.65	0.63

Table B.8: Passenger Statistics when HUB1 under IFR and HUB2, HUB3 under VFR

	D-FAM			D-ARM			Traditional
	Layer1	Layer2	Total	Layer1	Layer2	Total	Total
No. of Total Pax	43,349	21,674	65,023	40,955	24,068	65,023	65,023
No. of Pax miss connection	726	2,365	3,091	674	2,586	3,260	2,972
No. of Pax didn't get served	179	512	691	162	523	685	593
Avg Pax Delay (min)	10.5	33.3	18.1	10.1	32.5	18.4	17.8
Pr (on-time)	0.79	0.52	0.70	0.79	0.54	0.69	0.68
Revenue Lost	37,591.3	93,129.5	130,721	34,599.2	96,873.4	131,473	114,213
Operating Cost Saving	N/A	N/A	0	N/A	N/A	0	0
Delay Crew Cost	N/A	N/A	338,564	N/A	N/A	325,328	316,679
Total Cost Lost	N/A	N/A	469,285	N/A	N/A	456,801	430,892

Table B.9: Flight Statistics when HUB2 under IFR and HUB1, HUB3 under VFR

	D-FAM			D-ARM			Traditional
	Layer1	Layer2	Total	Layer1	Layer2	Total	Total
No. of Flights	680	454	1134	681	453	1134	1134
No. of Flights Cancelled	0	3	3	0	5	5	0
Avg Flight Delay (min)	11	21	15	10	23	15	15
StdDev Flight Delay (min)	18	35	27	17	35	27	23
Pr (on-time)	0.76	0.63	0.71	0.76	0.62	0.70	0.68

Table B.10: Passenger Statistics when HUB2 under IFR and HUB1, HUB3 under VFR

	D-FAM			D-ARM			Traditional
	Layer1	Layer2	Total	Layer1	Layer2	Total	Total
No. of Total Pax	43,349	21,674	65,023	40,955	24,068	65,023	65,023
No. of Pax miss connection	525	1,105	1,630	441	1,312	1,754	1,588
No. of Pax didn't get served	142	243	385	121	213	334	331
Avg Pax Delay (min)	10.9	23.2	15.1	10.0	27.6	16.5	16.0
Pr (on-time)	0.77	0.67	0.74	0.78	0.63	0.72	0.69
Revenue Lost	31,253.6	45,880.5	77,134.1	27,068.8	40,910.8	67,979.7	67,759.7
Operating Cost Saving	N/A	N/A	11,737.2	N/A	N/A	11,856.7	0
Delay Crew Cost	N/A	N/A	291,355	N/A	N/A	295,341	286,080
Total Cost Lost	N/A	N/A	356,751.9	N/A	N/A	351,464	353,839.7

Table B.11: Flight Statistics when HUB3 under IFR and HUB1, HUB2 under VFR

	D-FAM			D-ARM			Traditional
	Layer1	Layer2	Total	Layer1	Layer2	Total	Total
No. of Flights	680	454	1134	681	453	1134	1134
No. of Flights Cancelled	0	10	10	0	3	3	0
Avg Flight Delay (min)	11	16	13	10	17	13	13
StdDev Flight Delay (min)	19	29	24	18	28	23	22
Pr (on-time)	0.77	0.71	0.75	0.78	0.68	0.74	0.74

Table B.12: Passenger Statistics when HUB3 under IFR and HUB1, HUB2 under VFR

	D-FAM			D-ARM			Traditional
	Layer1	Layer2	Total	Layer1	Layer2	Total	Total
No. of Total Pax	43,349	21,674	65,023	40,955	24,068	65,023	65,023
No. of Pax miss connection	550	1,327	1,877	431	1,235	1,666	1,575
No. of Pax didn't get served	143	223	366	117	200	317	331
Avg Pax Delay (min)	10.7	22.6	14.6	9.7	21.8	14.2	13.6
Pr (on-time)	0.79	0.72	0.76	0.80	0.68	0.76	0.75
Revenue Lost	31,101.9	42,430.7	73,532.6	26,174.2	39,488.4	65,662.6	68,881
Operating Cost Saving	N/A	N/A	30,392	N/A	N/A	21,510.6	0
Delay Crew Cost	N/A	N/A	241,951	N/A	N/A	247,381	243,175
Total Cost Lost	N/A	N/A	285,091.6	N/A	N/A	291,533	312,056

Bibliography

[1] U. S. major's top 10 domestic airports, November 2000. Aviation Daily.

[2] J. Abara. Applying integer linear programming to the fleet assignment problem. *Interfaces*, 19:20–28, 1989.

[3] Y. Ageeva. Approaches to incorporating robustness into airline scheduling. Master's thesis, Massachusetts Institutte of Technology, 2000.

[4] M. Arguello, J. Bard, and G. Yu. A grasp for aircraft routing in response to grounding and delay. *Journal of Combinatorial Optimization*, 1(3):211=228, 1997.

[5] Federal Aviation Association. Airport capacity benchmark report 2001, 2001. http://www.faa.gov/events/benchmarks.

[6] J. Bard, G. Yu, and M. Arguello. Optimizing aircraft routings in response to groundings and delays. *IIE Transactions*, 33:931–947, 2001.

[7] C. Barnhart, N. Boland, L. Clarke, E. Johnson, G. Nemhauser, and R. Shenoi. Flight string models for aircraft fleeting and routing. *Transportation Science*, 32(3):208–220, 1998.

[8] C. Barnhart, C. Hane, and P. Vance. Using branch-and-price-and-cut to solve origin-destination integer multicommodity flow problems. *Operations Research*, 48(2):318–326, 2000.

[9] C. Barnhart, E. Johnson, G. Nemhauser, M. Savelsbergh, and P. Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46(3):316–329, 1998.

[10] C. Barnhart, T. Kniker, and M. Lohatepanont. Itinerary-based airline fleet assignment. *Transportation Science*, 36(2):199–217, 2002.

[11] M. Bartholomew-Biggs, S. Parkhurst, and S. Wilson. Using direct to solve and aircraft routing problem. *Computatinoal Optimization and Applications*, 21:311–323, 2002.

[12] M. Berge and C. Hopperstad. Demand driven dispatch: A method for dynamic aircraft capacity assignment, models and algorithms. *Operations Research*, 41(1):153–168, 1993.

[13] D. Bertsimas and S. Patterson. The air traffic flow management problem with enroute capacities. *Operations Research*, 46(3):406–422, 1998.

[14] D. Bertsimas and J. Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, 1997.

[15] S. Bratu. *Airline Passenger On-Time Schedule Reliability: Analysis, Algorithms, and Optimization Decisions Models*. PhD thesis, Massachusetts Institutte of Technology, 2004.

[16] K. Budenbender, T. Grunert, and H. Sebastian. A hybrid tabu search/branch-and-bound algorithm for the direct flight network design problem. *Transportation Science*, 34(4), 2000.

[17] S. Chebalov and D. Klabjan, editors. *Robust Airline Crew Scheduling: Move-up Crews*, Proceedings of the 2002 NSF Design, Service, and Manufacturing Grantees and Research Conference, 2002.

[18] M. Clakre. An introduction to the airline recovery problem. Master's thesis, Massachusetts Institutte of Technology, 1997.

[19] L. Clarke, C. Hane, E. Johnson, and G. Nemhauser. Maintenance and crew considerations in fleet management. *Transportation Science*, 30(3):249–260, 1996.

[20] A. Cohn. *Composite-Variable Modeling for Large-Scale Problems in Transportatoin and LOgistics.* PhD thesis, Massachusetts Institutte of Technology, 2002.

[21] A. Cohn and C. Barnhart. Improving crew scheduling by incorporating key maintenance routing decisions. *Operations Research*, 51(3):387–798, 2003.

[22] J. Cordeau, G. Stojkovic, F. Soumis, and J. Destosiers. Benders decompositoin for simultaneous aircraft routing and crew scheduling. *Transportation Science*, 35(4):375–388, 2001.

[23] G. Desaulniers, J. Destosiers, Y. Dumas, M. Solomon, and F. Soumis. Daily aircraft routing and scheduling. *Management Science*, 43(6):841–855, 1997.

[24] A. Evans. Responses to airport delays - a system study of newark international airport. Master's thesis, Massachusetts Institutte of Technology, 2002.

[25] J. Fitzsimmon and M. Fitzsimmon. *Service Management for Competitive Advantage.* McGraw-Hill Co., 1993.

[26] E. Fumagalli, J. Black, M. Ilic, and I. Vogelsang. A reliability insurance scheme for the electricity distribution grid. *IEEE*, 2001.

[27] B. Garcia, J. Potvin, and J. Rousseau. A parallel implementation of the tabu search heuristic for vehicle routing problems with time window constraints. *Computers Operations Research*, 21(9):1025–1033, 1994.

[28] M. Gendreau, F. Guertin, J. Potvin, and E. Taillard. Parallel tabu search for real-time vehicle routing and dispatching. *Transportation Science*, 33(4):381–390, 1999.

[29] F. Glover. Tabu search – part i. *ORSA Journal on Computing*, 1(3):190–206, 1989.

[30] F. Glover. Tabu search – part ii. *ORSA Journal on Computing*, 2(1):4–32, 1990.

[31] F. Glover. Tabu search: A tutorial. *Interfaces*, 20:74–97, 1990.

[32] F. Glover. A user's guide to tabu search. *Annals of Operations Research*, 41:3–28, 1993.

[33] F. Glover and M. Laguna. *Tabu Search.* Kluwer Academic publishers, 1997.

[34] R. Gopalan and K. Talluri. Mathematical models in airline schedule planning: A survey. *Annals of Operations Research*, 76:155–185, 1998.

[35] W. Hall. *Efficient Capacity Allocation in a Collaborative Air Transportation System.* PhD thesis, Massachusetts Institutte of Technology, 1999.

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

[37] I. Ioachim, J. Destosiers, F. Soumis, and N. Belanger. Fleet assignment and routing with scheudle synchronization constraints. *European Journal of Operations Research*, 119:75–90, 1999.

[38] J. Kelly and J. Xu. A set-partitioning-based heuristic for the vehicle routing problem. *INFORMS Journal on Computing*, 11(2):161–172, 1999.

[39] D. Klabjan, E. Johnson, and G. Nemhauser. Solving large airline crew scheduling problems: Random paring generation and strong branching. *Computational Optimization and Applications*, 20:73–91, 2001.

[40] D. Klabjan, E. Johnson, G. Nemhauser, E. Gelman, and S. Ramaswamy. Airline crew scheduling with time windows and plane-count constraints. *Transportation Science*, 36(3):337–348, 2002.

[41] D. Klabjan, A. Schaefer, E. Johnson, A. Kleywegt, and G. Nemhauser, editors. *Robust Airline Crew Scheduling*, Proceedings of TRISTAN IV, 2001.

[42] S. Kontogiorgis and S. Acharya. Us airways automates its weekend fleet assignment. *Interfaces*, 29:52–62, 1999.

[43] L. Lettovsky, E. Johnson, and G. Nemhauser. Airline crew recovery. *Transportation Science*, 34(4):337–348, 2000.

[44] M. Lohatepanont and C. Barnhart. Airline schedule planning: Integrated models and algorithms for schedule design and fleet assignment. *Transportation Science*, 2002. To appear.

[45] J. Milner. *Dynamic Slot Allocation with Airline Participation.* PhD thesis, Massachusetts Institutte of Technology, 1995.

[46] B. Rexing, C. Barnhart, and N. Krishnamurthy. Airline fleet assignment with time windows. *Transportation Science*. 34(1):1–20, 2000.

[47] J. Rosenberger, E. Johnson, and G. Nemhauser. Rerouting aircraft for airline recovery. *Transportation Science*, 2001. To appear.

[48] J. Rosenberger, E. Johnson, and G. Nemhauser. A robust fleet assignment model with hub isolation and short cycles. *Transportation Science*, 2001. To appear.

[49] R. Rushmeier and S. Kontogiorgis. Advanceds in the optimization of airline fleet assignment. *Transportation Science*, 31(2):159–169, 1997.

[50] H. Schildt. *C++ the Complete Refernece.* McGraw-Hill, 1995.

[51] R. Sedgewick. *Algorithms in C++*. Addison-Wesley, 1992.

[52] Z. Shavell. The effects of schedule disruptions on the economics of airline operations. 3rd FAA/Eurocontrol Air Traffic Management Research and Development Seminar, 2000.

[53] B. Thengvall, J. Bard, and G. Yu. Balancing user preferences for aircraft schedule recovery during irregular operations. *IIE Transactions*, 32:181–193, 2000.

[54] B. Thengvall, G. Yu, and J. Bard. Multiple fleet aircraft schedule recovery following hub closures. *Transportation Research Part A*, 35:289–308, 2001.

[55] S. Yan and Y. Tu. Multifleet routing and multistop flight scheduling for schedule pertubation. *European Journal of Operations Research*, 103:155–169, 1997.

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