Multi-Objective Airline Schedule Recovery

Yong Yean Yik @ Yong Yean Fatt

National University of Singapore
2005

Acknowledgements

I would like to use this opportunity to thank all the ISE staff that has rendered helps to me during the 2 year period stay in National University of Singapore. In particular I wish to thank Prof Huang for hiring me as a research engineer and giving me a unique opportunity to pursue studies in Industrial and Systems Engineering.

I also would like to thank Prof Lee Loo Hay and Dr Lee Chulung for being my supervisor and providing guidance to my master course. In particular the advice and guidance provided by Prof Lee has been most invaluable and I wish to express my sincerest thanks to him. I also would like to thank Prof Lee for putting up with the repeated delay in my thesis submission.

Finally I would like to thank all my friends and laboratory mates in NUS. You have all helped me to have an enjoyable stay in NUS.

Name: Yong Yean Yik @ Yong Yean Fatt

Degree: Master of Engineering

Dept: Department of Industrial & Systems Engineering

Thesis Title: Multi-Objective Airline Schedule Recovery

Airline schedule recovery in the airline industry involves decisions concerning aircraft reassignment where normal day to day airline operation is disrupted by unforeseen circumstances, such as bad weather conditions causing flight delays. Airline schedule recovery attempts to recover these flight schedules through a series of reassignment of aircrafts and readjustments of scheduled flying time.

Two mathematical models are proposed in this thesis in attempt to produce optimal airline schedule recovery solutions during a disruption event. The first model attempts to minimize passenger disrupted by such a reassignment while attempting to maximize their on-time percentage index. The constraints considered in this model include aircraft balance at each node in time-space network and passenger itineraries. The second model expands upon the first model by adding aircraft maintenance consideration into the first model.

The effectiveness of the models are tested using an airline schedule simulation software SimAir. Throughout the work presented here, the focus has been to develop methods which are simple, extendable and able to produce an optimal solution in a relatively short time.

Content Page

Chapter 1

Introduction

This chapter looks at the challenges of airlines facing in today's competitive market and establishes the importance of good recovery procedures. This in turn leads to the motivation of this thesis in using mathematical modeling to solve airline schedule recovery problem. Two mathematical models are proposed. The first attempts to recover a disrupted schedule by minimizing number of passengers disrupted and maximizing overall on time performance index. The second model expands upon the first by adding in aircraft maintenance consideration.

1.1 Background

The airline industry is becoming increasingly competitive. In some regions, South East Asia for example, there is increasing competitors entering into what is essentially an already highly competitive market. Several major events in the past few volatile years (increasing fuel prices, SARS epidemic, terror attacks) only serve to put more woes to an embattled industry. In addition to that airlines need to compete for customers against other modes of transport such as trains and buses.

For an airline to survive in such competitive environment they must be able to provide quality services. They must provide on-time services and subject their passengers to as little hassle as possible. To achieve that they must utilize their given resources as best as they could. When disruption occurs, airlines would want to return to the normal schedule as soon as possible.

The recent 2 years see the introduction of a number of low cost carrier airlines, especially in South East Asia. The competitive pricing of these budget airlines put increasing pressure on traditional airlines. To compete, traditional airlines need to revise their operations, reduce cost and improve services.

Airlines spend a great deal of effort to develop flight schedules for each of their fleet. A seasonal flight schedule is made up of a collection of flight legs. A flight leg typically consists of an originating station, departure time, a terminating station, and expected arrival time. Aircrafts are assigned to cover these flight legs so that each and every flight leg within the schedule is covered by one aircraft. A continuous series of flight legs a particular aircraft flies, form an aircraft route. Each aircraft, upon finishing a flight leg, would typically park at the gate of the destination airport for a certain amount of minutes. This is called turn time, and it is necessary for maintenance crews and cleaning crews to perform their duties on the idle aircraft while passengers gathered for the next flight prepares to board the aircraft at the gate. Minimum turn time refers to the least amount of time an aircraft must wait at the gate before serving the next flight. Typically minimum turn time varies from 30 minutes to 40 minutes. If an aircraft is scheduled to stay longer than the minimum turn time at the gate, the excess time translates into slack time for the airline.

Due to the high costs associated with the purchase and subsequent maintenance of these aircrafts, airlines attempt to maximize the usage of aircrafts as much as possible. This desire often translates into tightly coupled aircraft routings, with little to no slack time in between two consecutive legs. While sound in theory, the moment an aircraft is unexpectedly grounded or delayed (which happen almost daily), the lack of slack to compensate for it causes subsequent flights to be delayed as well.

1.2 Airline Schedule Disruption

Due to various unforeseen circumstances, airline schedules are almost always disrupted on a daily basis. The type of disruption encountered may be minor (a delay to departure for 5 to 10 minutes), or major (several aircrafts are grounded for hours).

There are various factors causing the disruption to an airline schedule. Occasionally, an aircraft needs to undergo unexpected maintenance checks. The maintenance crews, while performing routine checks, discovers degraded components/conditions in aircrafts and thus requires extra maintenance before it can service the next flight. Since these maintenances are not scheduled, they are typically called unscheduled maintenance. Depending on the magnitude of the problem, it may last anywhere between 30 minutes up to days on end. Naturally the flights that the aircraft is scheduled to fly would have to be delayed, or cancelled.

The amount of time spent on gate delays, duration for taxi into and taxi out of gates, actual flight duration and aircraft runway queuing time, are often modeled as stochastic processes. Various minor delays at these stages can and often do accumulate up resulting significant overall flight delays.

During peak hours, congestions at airports contribute significantly to aircraft delays. Bottlenecks often materialize at places of shared resources. For example, an aircraft may be held up in airspace queue or runway queue while waiting for its turn to utilize the runway. In some airports where gates are shared between different airlines, aircrafts often need to wait for its turn to utilize a gate that is currently occupied by a delayed flight. While taxiing in and out of gates, congestion on taxi ways may delay the aircraft's schedule even further.

Inclement weather condition is another major source of schedule disruption. Bad visibility during thunderstorm will mean aircrafts require longer runway occupation time and aircraft separation time in order to take off and land. Runway and airspace queues of aircrafts waiting their turn to use the runway would stack up, and in turn bring about even more delays. In extremely bad weather condition, such as a snow storm, runways are closed and aircrafts are grounded for an indeterminate period, until weather improves again. Obviously such delays have serious repercussions to airline schedules.

Living in the aftermath of September 11 incident (Harumi Ito 2003), with the recent spate of security breach incidents in some major US airports, entire airport is closed down and all aircrafts are grounded. Most flights are delayed up to 6 hours or more. This too constitutes a formidable challenge to airlines in operating with such major disruptions.

The tightly coupled airline schedule imply that a single disruption at one point in the aircraft schedule network will have repercussion ripple through down the network and be felt even hours later. A late arrival of a certain aircraft, in addition to causing delay to its next flight, may also impact other flights in the network. For example, there are instances where aircrafts are delayed from departing from gate even though it is ready to depart on time, because it has significant number of connecting passengers still trapped in a prior flight that is delayed.

Naturally an airline schedule disruption is considered by all parties a negative incident, both detrimental to airline's reputation and creates passenger inconveniences. Major disruptions are costly too. For example, disruptions would often mean reassignment to crew schedules, and such reassignment often incurs monetary penalties. Flight delay, and in some cases flight cancellations, result in loss of customer goodwill, and indirectly results in loss of eventual revenue.

Federal Aviation Administration (FAA) requires all major American airlines to make public their on time performance indexes. In a nutshell, on time performance index refers to the percentage of flights arriving no later than fifteen minutes after the scheduled arrival time, against all the scheduled flights over a period of that month. More specifically, on-time performance = 1-(number of flights arriving later than fifteen minutes of scheduled arrival time) divided by (total number of flights).

A low on time performance is considered bad publicity for the airlines, and major airlines are considerably concerned with this index. This index is used widely within the industry as a gauge on customer satisfaction. It is not uncommon that airlines opted to cancel a flight rather than delaying it, in view that the delay will degrade the performance index.

1.3 Airline Schedule Recovery

The operational decision on how to reschedule the flights is commonly called aircraft schedule recovery, and is a major source of headache for major American airlines these days. In general a recovery plan touches on several different aspects of operations, with multiple objectives, often conflicting with each other, to be considered. This is further compounded by the fact that airlines must solve the problem within a very short time interval whenever a disruption occurs.

There are various aspects of operational consideration impacting on airline schedule recovery. These include:

- the utilization of available fleet of aircrafts;
- re-accommodation of passengers affected by such changes;
- reassignment of crews to follow the new schedule; and
- liaising with airports involved regarding the gate re-assignment.

Naturally the impact of disruption must be contained, and not allowed to propagate on for too long. It must not cost the airline too much loss of sales with flight cancellations. Reassignment of crews should preferably be minimal as reassignment often costs extra charges. Flight delays should hopefully be minimal, or the overall on-time performance of the airline would suffer, impacting the reputation of airline negatively. Given so many objectives to balance, some of which actually in conflict with one another, it is clear a mathematical model is needed to solve such a complex problem.

In addition to the various objectives stated above, there are a few other considerations an airline schedule recovery planner must consider.

Firstly, aircraft balance must be maintained. If the flight leg f_1 of aircraft a_1 to station s_1 is cancelled, there must be a spare aircraft at station s_1 to fly the flight leg f_2 that a_1 is scheduled to fly originally. If no idle aircrafts are around to fly the subsequent flight leg f_2, then one must either cancel f_2 or delay it indefinitely, until a spare idle aircraft is around to service it. In other words, the number of flight legs flowing out of a station, minus the number of flight legs flowing into a station, over a period of duration, must equal the number of idle aircrafts originally at the station.

Secondly, aircrafts must meet their maintenance requirements. A recovered schedule that assigns the flight routes to all the aircrafts must not cause the aircrafts to violate their maintenance requirements. In general each aircraft needs to undergo scheduled maintenance every three to four days, and these maintenances are only provided at certain station. If, at the point of disruption, an aircraft must undergo maintenance within the next 24 hours, the recovered schedule must make provision such that maintenance is scheduled for that aircraft. Maintenance consideration is tackled in the second model in this research.

Thirdly, there must be an end to the recovery period. Naturally there must be a cut off to schedule changes, beyond which the flight schedule resumes as per-normal. In most cases, flight schedules are allowed to be changed, starting from point of disruption, to the end of the day. Flights will resume as per normal the next day. In more serious disruption, the end of recovery period is extended into the following day.

There are several options a schedule recovery planner can consider in order to bring flight schedule back to normalcy.

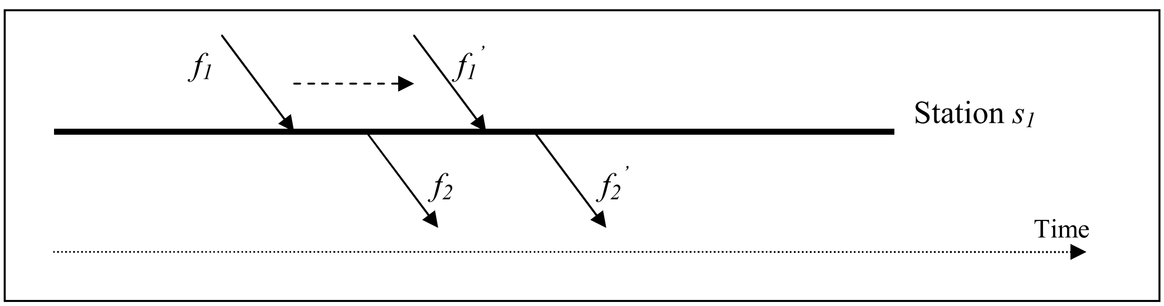

Figure 1.1: Flight Delay

In many cases a simple delay of flight would be sufficient. In the above example flight f_2 is delayed to a later time in order to connect with a delayed flight f_1.

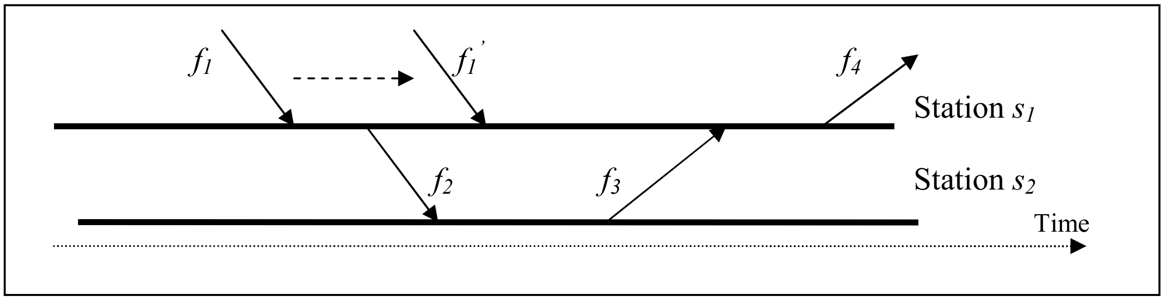

Figure 1.2: Flight Cancellation

In other cases one may need to consider canceling the disrupted flight, provided the benefit of doing so offsets the loss of revenue and passenger goodwill on the cancelled flight. Using the above example, one may choose to cancel flight f_2 and f_3 so that the delayed flight f_1' can resume its flight with flight f_4.

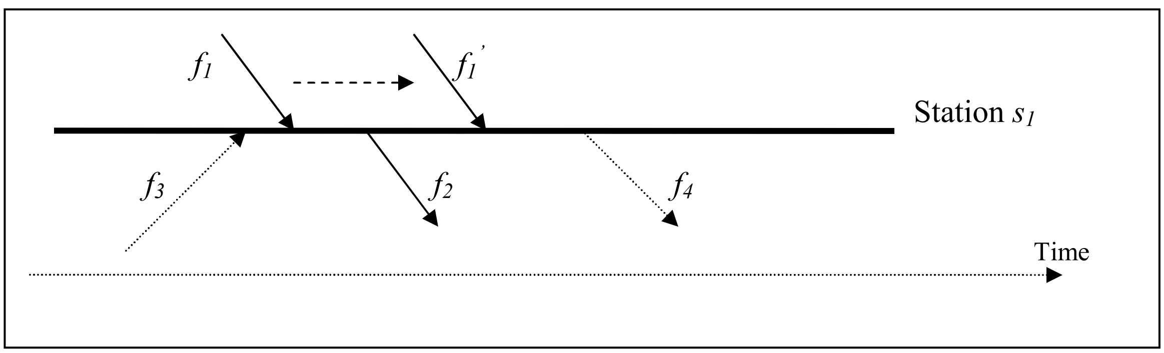

Figure 1.3: Aircraft Swapping

One may opt to swap the aircrafts scheduled for different flight. Using the above example, aircraft a_1 is scheduled to fly flight leg f_1, but f_1 is expected to arrive at a later time f_1', and this would cause aircraft a_1 to miss the scheduled departure time of f_2. In the mean time aircraft a_2, having arrived via an earlier flight f_3, is scheduled to serve flight f_4 later. A simple swapping of flight leg assignment, assigning aircraft a_2 to fly f_2 on-time, while aircraft a_1 fly f_4 later, would incur no schedule delay.

In more complex instances, a combination of cancellation, delays and swapping may be necessary.

Given that this is a multi-objective problem with constrained resources, there is an optimal way to rearrange the schedule so that the disruption effect is minimized. It is a sufficiently challenging problem since often there are multiple operational issues that must be taken into consideration during a recovery. In addition the number of flight legs involved is considerable, often numbering in the hundreds, which means heuristics and rule-of-thumb employed by human decision maker would not yield an optimal solution.

1.4 Research Objective

It is noteworthy that few of the airline schedule recovery algorithms utilized by airlines currently provide satisfactory solutions. To the best of our knowledge, none of the current literatures on airline schedule recovery, manage to capture all the myriad operational issues plaguing a real life schedule recovery problem. While theoretically it is possible to model every single real life constraints into a mathematical model, it often renders the whole problem into an NP-hard problem that may take more than a few hours to arrive at an optimal solution. This, however, is not acceptable since in real life, recovered schedule must be generated in a sufficiently short time, often in the matter of minutes.

This research is not an attempt to come up with an all embracing model managing to capture all the possible operational considerations. Instead, attempts are made to look at operational constraints that are often neglected or overlooked in other literature.

There has been little effort spent on tackling recovery from the passenger point of view. Also, there are very few papers that tackle aircraft maintenance consideration. Aircraft maintenance, if at all considered, is solved on the side instead of being an integral part of the mathematical model.

As far as we know, few of the papers looked at on-time performance as an objective criterion. Almost all the mathematical models attempt to minimize overall delay. For an airline very much concerned with on-time performance, a 16 minute delay is no different from a 20 minute delay since both delays are already later than the 15 minutes delay used in calculating on-time performance. This is contrasted with minimize overall delay that is used in almost all the papers cited in previous chapter.

In this research we focus on 2 major issues that concern an airline during a recovery operation: how to best reschedule the aircrafts, minimizing disruption to passengers, while at the same time keep up the on-time performance index of the airline?

Passenger itineraries are disrupted whenever the flight legs in the itinerary are cancelled. When this occurs, passengers following these itineraries would not be able to reach their intended destination. In addition, for passenger itineraries featuring more than 1 leg, misconnection can occur whereby the former incoming flight leg arrives later than the departure of the latter outgoing flight leg. In these situations, the passengers on board of the former flight leg would fail to catch the latter flight leg. In this research we shall consistently refer to inconvenienced passengers as passengers that fall into the above 2 categories. More precisely, inconvenienced passengers are passengers that would fail to reach their intended destination via the planned itinerary, due to either

- Cancellation of any one of their flight legs in the itinerary
- Misconnection where there is insufficient connection time for passengers in the former incoming flight to connect to the latter departing flight.

In our model passengers are modeled in unit of passenger itineraries. A passenger itinerary is defined as a group of passengers from the same origin station and intended to reach the same destination. Their scheduled series of flight legs that would lead them from origin to destination are exactly the same.

In most cases, passenger itineraries consist of just one flight leg. Passengers flying on these itineraries will be inconvenienced when the flight leg is cancelled. In other cases, a passenger itinerary may consist of two or more flight legs. For example, a flight from Singapore to cities in the United States generally requires a stop-over in Narita airport, and possibly another stop at San Francisco. Passengers in these passenger itineraries are prone to disruption. In addition to the possibility of flight cancellation of any of these legs, they may also encounter misconnection: situation where passenger turn time between incoming flight and outgoing flight during a connection is insufficient.

The recovery model proposed in this research addresses the problem of aircraft recovery from a multi-objective point of view. The objective of the models proposed is to minimize number of passengers inconvenienced/disrupted balanced against on time performance of airline. The approach of handling multi-objective problem is elaborated more throughout below.

It is clear that the two objectives stated above are mutually opposing objectives. On the one extreme, one may simply delay all the subsequent flight legs to make sure no passenger misses their connection due to misconnection. On the other extreme, one may simply cancel a lot of flights, creating a lot of slack time in the schedule just so that all flights can depart exactly on time.

The two opposing objectives are linked into a single objective, with their corresponding objective weights be varied through a large number of simulation runs. The end result of the sets of simulation runs would form a set of pareto optimal solutions that airline companies can pick and choose.

An additional objective of this work is to produce a solution that can yield a solution within a reasonably short time. The resultant algorithm must provide a satisfactory solution to a reasonably large sized problem within reasonable time when the disruption occurs. A second model that is an extended version of the first, is also proposed. In this second recovery model, aircraft maintenance needs are also considered. In practice, a manual rescheduling takes the airline planners up to half a day, so there is much value added if a recovery plan can be generated with computer aid within a much shorter time frame.

Due to the highly stochastic nature of the problem, an airline operation simulation software is coded and used to validate the result of the proposed mathematical recovery model, The simulation result of the two proposed models is compared against each other. In addition, a set of default heuristic recovery rules is also simulated and compared against the two recovery models.

1.5 Outline of Thesis

The following chapter examines the existing airline schedule recovery algorithms presently published. This would lead to justification of the focus of the research detailed in this thesis.

Chapter 3 details the first mathematical model proposed in this work, which is a multi-objective recovery model with passenger disrupted and on-time performance index consideration. The model attempts to do so by minimizing the overall passenger disrupted and maximizing the on-time performance index of the airline.

Chapter 4 details a second mathematical model, which is an extended version of the model proposed in chapter 3. This second model takes aircraft maintenance requirements into consideration on top of the considerations stated above.

Chapter 5 introduces SimAir: A discrete event simulation software that simulates airline operation. SimAir is used extensively in this research. The 2 mathematical models are incorporated into SimAir to simulate airline operations of a major American airline, and various data statistics are collected over a simulated duration of 7 day airline schedule. In addition to the 2 mathematical model recoveries, SimAir has its own default recovery procedure, utilizing a set of heuristic rules to handle airline schedule recovery.

The simulation results from the two proposed mathematical models are compared against the default existing airline schedule recovery in SimAir. The results and comparisons for the 2 models are detailed in chapter 6 and chapter 7 respectively.

Chapter 8 concludes the thesis with some discussion on the finding of this research work.

Chapter 2

Literature Review

This chapter lists out relevant journal and publications related to airline schedule recovery performed over the years. It attempts to list out significant contributions that lead airline schedule recovery algorithms to current state. The chapter would also highlight the (thus far) lack of academic attention on passenger recovery and airline maintenance consideration, which in turn motivated this research.

2.1 Literature Review

Given that airline schedule recovery problem is a rather complex problem to be solved; there is no lack of published papers that attempt to address the problem of airline schedule recovery.

In the paper published by Teodorovic and Guberinic (1984), it proposes a branch-and-bound procedure in the search of an optimal solution that minimizes total passenger delay. The work does not document results of solving problems of a realistic size. Instead a solution solving a simple example of 8 flights is provided.

Teodorovic (1985) presents research on the reliability of airline scheduling as it relates to meteorological conditions, the ability to identify an indicator for qualifying the adaptability of such airline schedules to weather conditions. The author outlines a heuristic algorithm for minimizing the number of aircrafts required to satisfy a set of traffic volume.

Theodorovic and Stojkovic (1990) solve the schedule recovery problem by formulating a model with two objectives using lexicographic optimization. The primary objective maximizes the number of flights flown while the secondary objective minimizes total passenger delay. Flight links for each aircraft are created via a greedy heuristic. These links are then solved using a shortest path problem for each aircraft where the arc cost carries the primary and secondary objectives. It is found that the model is highly sensitive to how the objectives are ranked.

Jarrah et al. (1993) propose a model that allows swapping of aircrafts of the same equipment type. The underlying solution methodology is based on network flow theory. Delays are allowed for departing flights to compensate for aircraft shortages at a given station. The model is solved as a minimum cost network problem.

Yan and Young (1996) apparently are the first to propose a model that allows delays and cancellations to be considered simultaneously. The objective of their model is to maximize the profit of the airline. The problem is solved as a minimum cost flow problem with side constraints. It is solved using Lagrangian relaxation with a sub-gradient method. They outline the basic schedule perturbation model which is designed to minimize the schedule perturbed period after an incident. Maintenance schedules and passenger connections are ignored here.

Cao and Kanafani (1997a), whose work can be viewed as an extension of Jarrah's work (1993) above, discuss a real-time decision support tool for the integration of flight cancellations and delays. The paper presents a 0-1 quadratic programming model, which maximizes airline profit while taking into consideration delay costs and penalties for flight cancellations. Special properties of their Flight Operations Decision Problem (FODP) model are exploited to allow a specialized algorithm to solve the problem in real time. The model considers delays and aircraft reassignments from one station to the next. The author extended upon their base model to incorporate aircraft ferrying and multiple aircraft type swapping capabilities.

Cao and Kanafani (1997b) present in their subsequent article an effective algorithm to solve the FODP model. In this paper, they discuss the computational result of a continuous mathematical problem, derived from their 0-1 quadratic problem. Unfortunately, their case studies do not consider aircraft ferrying, crew scheduling and airport capacity.

Arguello et al. (1997) considers an airline schedule recovery problem in the event aircraft gets grounded or delayed. The goal here is to produce a recovered schedule that lasts to the end of the day, and able to resume the normal schedule the following day. The objective is to minimize the costs that includes passenger inconveniences and lost flight revenues. The solution proposed is a "neighborhood search technique" heuristic that incorporates the basic component of GRASP (greedy randomized adaptive search procedure). Initial incumbent solution is found by canceling all flights that are to be flown by disrupted aircraft. By making minor changes to this incumbent, neighborhood solutions are created. Each of these neighborhood solutions are costed, and a restricted candidate list is used to keep the best costed solutions. For each subsequent iteration a new incumbent solution is picked randomly from the restricted candidate list. The terminating condition is either an empty restricted candidate list is encountered, or time limit is up.

The computational experiments reported by Arguello et al. are within the framework of a fleet of 16 aircrafts flying 42 flights with 13 airports in total. The heuristics typically find reasonably good solutions (within 10% of the optimality) within reasonably short processing time (10 seconds).

Talluri (1997) describes an algorithm that allows aircraft swaps without affecting equipment type composition of over-night aircraft at various stations within the airline's network. The algorithm is essentially a shortest path algorithm.

Lettovsky's Ph.D. thesis (1997) provided a model to solve integrated crew and aircraft schedule recovery problem. The thesis presents a linear mixed-integer mathematical problem that maximizes profit while capturing the availability of aircraft, crew and passengers. It turns out the problem is intractable for a reasonably sized scenario; hence a decomposition scheme is adopted instead. The master problem provides cancellation and delay options that satisfy landing restrictions.

Three sub-problems are then formulated. Aircraft Recovery Model (ARM), Crew Recovery Model (CRM) and Passenger Flow Model (PFM) each tackle the aircraft, crew and passenger consideration. Bender's decomposition is applied to the model to test the validity of the algorithm. SRM determines a plan for cancellation, delays and equipment assignment considering landing restrictions. For each of equipment type f, the model solves ARM_f. For each crew group c, the model solves CRM_c. These subproblems returns Benders feasibility or optimality custs to the SRM. Finally, PFM evaluates the passenger flow. The framework attempts to produce passenger friendly solution by adding Bender's optimality cut from PFM, while considering feasibility cuts from ARM_f and CRM_c. There is no computational experiment to demonstrate that larger problems can be solved within a reasonable time.

Yan and Lin (1997) applied network flow technique to develop several models to help airlines to handle temporary closure of airports. These models minimize the schedule-perturbed time after incidents so that airlines can resume their services as soon as possible. The models fall under network flow model or network flow model with side constraints. Network simplex method was employed to solve network flow model while Lagrangian relaxation-based solution algorithm is devised to handle network flow model with side constraints. The computational results show that in a reasonably sized problem (1773 nodes and 6860 arcs), solution are obtained within 1 minutes most of the time.

In the work by Yan and Tu (1997), they consider a recovery problem with multifleet and multistop flights. The framework is based on a basic multifleet schedule perturbation model (BMSPM) constructed as a time-space network from which strategic models are developed for incident scheduling. The resultant integer multiple commodity network flow problems are characterized as NP-complete problems. The paper proposes using Lagrangian relaxation with subgradient methods for approximating near optimal solutions. In the case studies provided, most models converge to 1% within at most half an hour of CPU time.

The paper by Benjamin Thengvall et al. (2000) also models the aircraft recovery problem as a network flow problem with side constraints. Several additional options were proposed as extension to their previous works. The proposed model models the schedule as a space-time network with flight arcs (various delay options of flights), ground arcs and nodes (termination and origination of various flight arcs). The proposed model allows options for delays and cancellations. It also incorporates a measure of deviation from the original aircraft routings, responding to Taiwan airline's request to produce a solution that does not deviate from the original schedule too much. Passenger connections and maintenance consideration are not considered in this model.

Benjamin Thengyall et al. (2003) continued their work on aircraft recovery problem in this more recent paper. A bundle algorithm is presented to solve a multi-commodity network model for determining a recovery plan for a single carrier with multiple fleets following a hub closure. A bundle algorithm is an extension of traditional sub gradient in which past information is used collectively to find the current search direction. The full methodology includes heuristics for finding feasible solutions from the solutions of the relaxed problems. On average, for larger test cases, a feasible solution is found twice as fast as a standard commercial code. The paper also claimed that it is able to generate several near optimal solutions. This is a plus since a number of practical constraints are generally not embedded into the model. Having multiple solutions provide a degree of flexibility to the airline recovery crew.

Stojkovic and Soumis (2001) addresses the problem of crew and flight schedule perturbation problem by modifying the planned duties for a set of available pilots to cover a set of flights by delaying (if necessary) some of these flights. Some flights will have fixed departure time while others will have more flexibility through a flight departure window. Stand-by pilots at stations are also modeled. They model the problem as a connection problem with explicit representation of each pilot. This renders the problem into an integer non-linear multi-commodity flow problem with additional constraints. The solution is approached by using column generation method, with master problem and sub-problem per-pilot. Three schedules, the largest of which includes 59 pilots and 190 flights, were solved and presented. The computational results were compared to traditional manual recovery.

Micheal Love et al. (2002) present another recent work on airline schedule recovery using heuristics. It is argued here that airline recovery algorithm must be able to provide a solution within a very short time. A problem size of 500 flights with 100 aircrafts must be solved in the interval of 3 minutes. It is argued that this stringent requirement requires them to employ heuristics instead. The paper focuses on rescheduling of aircrafts using simple search algorithms, with no consideration to crew and maintenance considerations. The objective value for each solution is calculated by tracking each aircraft through its link and calculates their contribution to the overall objective. In this work several search heuristic are tested, namely Iterated Local Search (ILS), and Steepest Ascent Local Search (SALS) and Repeated SALS (RSALS). It turns out SALS provides the most satisfactory solution overall. However due to much simplification quite a few considerations such as passenger connection and maintenance consideration are not considered here. It is also doubtful the quality of the solution given that there is no comparison made with regards to optimality solution.

Bratu et al. (2002) present two models that consider both aircraft, crew and passenger recovery. The basic model is a flight schedule network model. Flight delays are represented by several flight arcs. Both models do not consider how to recover disrupted crews. In their first model, Passenger Delay Model (PDM), delay costs are modeled more exactly than their second model, Disrupted Passenger Metric (DPM). A flight schedule involving 302 aircrafts, spanning over 74 airports, involving 9925 passenger itineraries is used for simulation and testing. Execution time ranges from 201 to 5042 seconds. The excessive execution time of PDM renders the model unfit for operational use. One noteworthy observation is, the passenger models presented in this thesis is exactly of the same form used in DPM model. However these models are developed without prior knowledge of each other.

Tobias Andersson et al. (2004) model the aircraft recovery problem as a multi-commodity flow problem with side constraints. One can treat the model as a mixed integer multi-commodity flow formulation where each aircraft is a commodity. Side constraints are used to model possible delays. The model allows delay and cancellation of flights, as well as aircraft swaps. The thesis contains 3 solutions for the problem: a Lagrangian heuristic, Dantzig-Wolfe method, and a tabu search method. Computational results are based on data from Swedish domestic airline (13-30 aircrafts, 2-5 fleet types, 98-215 flights, 19-32 airports). The Lagrangian heuristic results were not published, while the Dantzig-Wolfe method produced reasonable results for small sized problem. As problem size grows, the method use as much as 1100 seconds. The Tabu Search method consistently takes 10 seconds.

In a recently published technical report, Niklas Kohl et. al. (2004) provided a good round up of recent literature reviews on airline schedule recovery algorithms. In this paper it mentions that the passenger disruption model has not received much attention in the academic research at all. This could reflect the "old time" thinking where passengers only interesting when crew and aircraft is available. Also, in this paper it is mentioned that "..Due to complexity of the disruption management, there is little reason to believe it can be automated to the same extent as e.g. crew and fleet scheduling in the foreseeable future". This view is again supported in a separate technical report by Jens Clausen et al. (2005) which includes a good overview of literature review of recovery algorithms.

In almost all of the work cited above, passenger flow considerations and aircraft schedule maintenance are ignored. In some cases these constraints are added, but the resultant algorithm turns out to be intractable. In others, heuristics are employed, in hope of producing a solution in a much shorter time.

2.2 Discussion on Literature Reviewed

As evidenced by the abundance of papers in previous section, the problem of aircraft schedule recovery has received a lot of academic attention.

Unfortunately, none has yet managed to address this problem in a satisfactory manner, and it remains a challenging problem to be tackled.

Very few papers managed to capture the myriad operational issues plaguing a real life schedule recovery problem. Constraints such as crew availability, aircraft maintenance, airport departure slot limitation, aircraft balance, and fleet compatibility have so far not been captured in its entirety in any of the papers discussed above. Most tackled it from point of incorporating one or two constraints mentioned above. To handle all would result in an unwieldy collection of constraints that are NP hard to solve.

This is further compounded by the recovered schedule must be generated in a sufficiently short time, often in the matter of minutes. An overly complex mathematical model that attempt to tackle too many constraints at once, and takes more than 30 minutes to solve, would be useless from a practical point of view.

Interestingly, there are very few papers that tackle recovery from the passenger point of view. Also, there are very few papers that tackle maintenance consideration. Aircraft maintenance, if at all considered, is solved on the side instead of being an integral part of the mathematical model. None of the papers looked at on-time performance as a criteria: most attempt to minimize overall delay.

Chapter 3

Airline Schedule Recovery (Minimize Passenger Disruption and Maximize On-time Performance)

This chapter details the first recovery model. The chapter starts by stating the assumption of the model. It then explains the variables and indexes used throughout the rest of the chapter. Finally it states the recovery model itself.

3.1 Assumptions of Model

- The airline schedule recovery model proposed assumes there is a distinct beginning and ending time to the recovery process. In almost all instances, the beginning time of recovery occurs when disruption occurs. It is assumed that airline policy would dictate specific recovery duration, after which, the airline schedules must resume the normal operation. The time of end recovery is drawn by adding time of start recovery to this duration.

- All the flights set to occur in between the 2 timelines are involved in the recovery process. Each of these flights may encounter delays, aircraft swaps and even cancellations.
- The model does not make distinction between different fleet types of the airlines. It is assumed that all the flights involved in recovery can be serviced equally well by all the aircrafts involved in recovery.
- Crew constraints are not captured in this research.
- It is also assumed that the passenger itinerary data are readily available during a recovery process. This is essential since the proposed model requires passenger data to make decisions.
- The model assumes it is sufficient to ensure the flow balance of aircraft routes by the end of recovery period, but it does not require a specific aircraft to finally land at the originally designated station. In other words, aircraft A, originally designated to land at station X, may finally land at station Y while aircraft B, originally designated to land at station Y, may finally land at station X.

The next subsection is devoted to explaining the key indexes and variables used in this model, before the actual whole model is presented.

3.2 Variables and Indexes

For the purpose of recovery a mathematical formulation that utilizes binary variable flight f_{i,t_i} and binary connection variable x_{i,t_i,j,t_j} are used extensively. Flight variables f_{i,t_i} deal with decision to either delay a flight, or cancel it altogether. Connection variable x_{i,t_i,j,t_j} deals with decision of which next flight a particular aircraft, after serving the former, should serve next.

In addition, the beginning (*b*) and ending positions (*e*) are set up to indicate respectively the initial and final positions where the aircrafts should be. For the purpose of modeling passenger, passenger itinerary variable λ_p is created.

The following 4 subsections are each devoted to explaining these variables in greater detail.

3.2.1 Beginning and Ending positions

At the moment of the disruption, a timeline (recovery start time) is drawn cutting across the current flight schedule. The physical positions where the various aircrafts are currently at are noted.

The particular station and ready time (earliest time when an aircraft is ready to serve the next flight) of an aircraft, is termed beginning position (*b*) in the model. In the case of aircrafts that are currently not ready (currently still in the air or under maintenance) the expected arrival stations and expected available times of these aircrafts are set as the beginning positions instead.

A beginning position *b* has

1. a specific aircraft,
2. aircraft ready time, and
3. station where the aircraft is at

associated with it.

The collection of all beginning positions *b* form the set **B**.

A particular time span is set as the required period for the recovery process. The recovery duration is dictated by user and usually varies between half a day up to one day long. In more extreme scenarios of disruption longer recovery durations may be necessary. The moment when recovery process is completed is termed recovery end time.

This recovery end time line is drawn across the existing flight schedule and all the flight legs immediately beyond this time line are set as the end positions e of this current recovery.

All the aircrafts involved in the recovery process must rejoin back to the end positions dictated here, so that they may resume their normal flight schedule after the end of recovery process.

An ending position e has

1. aircraft termination time, and
2. station where the aircraft should end its route

associated with it.

The collection of all ending positions e forms the set **E**.

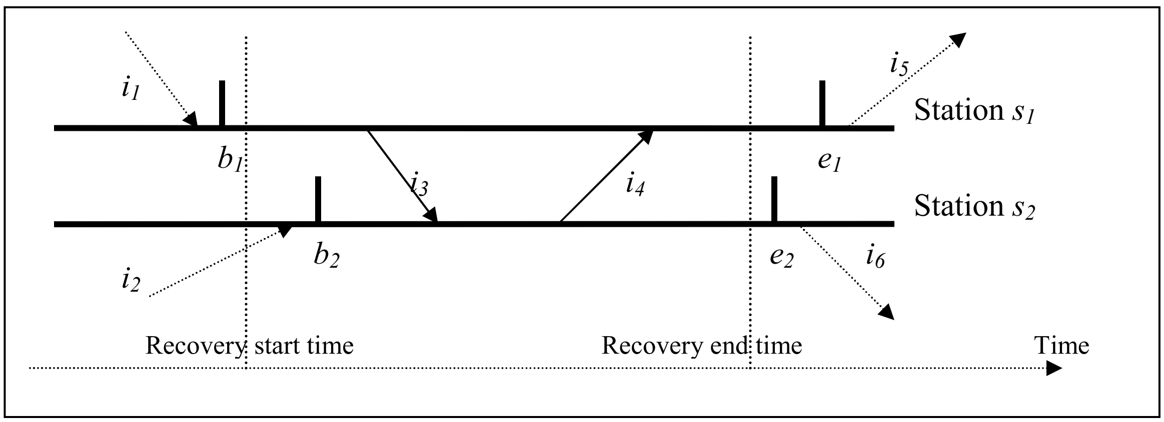

Figure 3.1: A Flight Plan Schematic

A flight plan schematic is shown above. Each bold horizontal line indicates a location, and each arrowhead indicates the scheduled departure and arrival time of each scheduled flight i_i across the horizontal time axis.

The two vertical lines serve as boundaries of the model to be solved. Only flights scheduled to depart after recovery start time, and flights scheduled to depart before the recovery end time, are involved in recovery.

Using the above example, b_1 and b_2 are respectively the beginning positions for the 2 aircrafts serving flight i_1 and flight i_2 at the moment of disruption. There is a small time gap between the arrival time of flight i_1 (i_2) and b_1 (b_2) because there is a minimum turn time required before aircraft may serve the next flight. Essentially, beginning positions refer to the time when various aircrafts are ready to serve any legs in the recovery.

In a similar manner, ending positions are created such that all the flights must eventually terminate at these various positions so that flight i_5 and i_6, which fall outside the end timeline, may be served by these aircrafts resuming the normal schedule.

3.2.2 Flight Variables

After determining the beginning positions and ending positions of a recovery scenario, all the flights falling within the 2 timeline will be involved in the recovery procedure. These flight legs are gathered into set **F**. Flight legs in set **F** are subjected to possible delay and even cancellation consideration, in order to bring flight schedule back to normal before the recovery end time.

To keep track of various delay/cancellation options of a flight *i*, binary variable f_{i,t_i} is introduced. Associated with each flight *i* is different t_i, denoting different delay options available to flight *i*. In particular, $f_{i,0}$ denotes flight i at its original departure time. If f_{i,t_i} =1 for a particular (i, t_i), it implies flight *i* chooses to depart at delay time t_i.

In addition, a binary cancellation variable $f_{i,c}$ is also created to allow the option of canceling flight *i*. If $f_{i,c}$ =1, it implies flight *i* is cancelled.

All flights are considered for cancellation. Delay in departure time for each flight is also considered. Depending on situation, different duration and number of flight delay options are generated for each of these flights.

Instead of generating flight delay options statically (that are spaced evenly from each other), a more intelligent algorithm is proposed. The algorithm detailing how these flight delay options are created is detailed in the following section.

3.2.3 Algorithm to Generate Flight Delay Options Dynamically

Although one may opt for a static manner of flight delay option generation (each flight leg spaced evenly 10 minutes apart, for example), in many cases they would create redundant variables. In this work a more dynamic generation of flight leg delay version is devised. Suppose a flight, upon delayed x amount of duration, does not make any new aircraft connection when compared to the original flight, then there is no potential benefit gained from such a delay. In other words, flight delays are only worthwhile if they present new aircraft connection possibilities.

This method helps in reducing the number of integer flight variables in the model, avoiding redundant versions of flight legs.

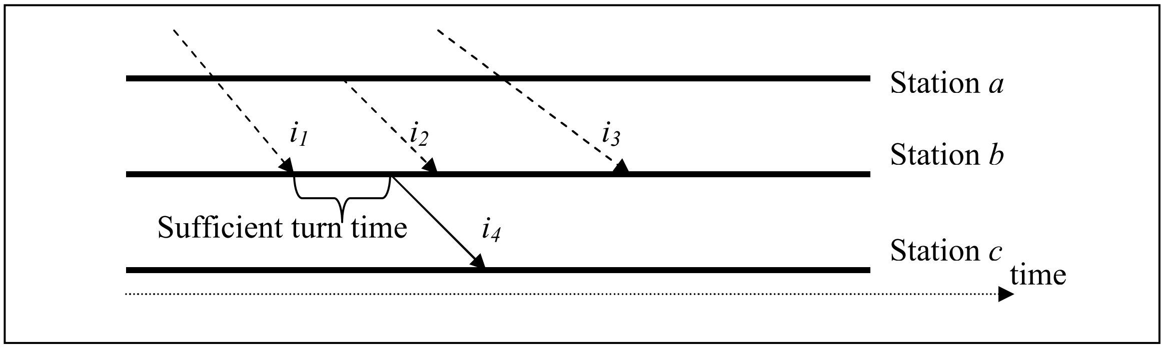

Figure 3.2: Flight Arrival Diagram

In the station-time diagram above, the arrowheads represent flight legs i_1, i_2, i_3 and i_4. The first 3 flight legs arrive at station *b*, which happens to be departure station of flight leg i_4. As indicated from the diagram, there is sufficiently large elapsed time between the arrival time of i_1 and i_4, such that an aircraft serving flight i_1 may choose to serve i_4 next. In contrast, there is insufficient elapsed time between arrival time of i_2 and departure time of i_4, thus any aircraft connection between i_2 and i_4 is not possible. Likewise, aircraft serving i_3 will not be able to "catch" i_4 and serve it next.

For the purpose of the following algorithm, $F_{Miss(i)}$ is defined to be the set of flight legs that have the same arrival station as the departure station of flight *i*, yet an aircraft connection between flight *i* and legs in $F_{Miss(i)}$ is impossible due to insufficient turn time. In particular, *justMiss(i)* is defined to be the flight leg that has the earliest arrival time among the set $F_{Miss(i)}$. Using the example above, i_2 and i_3 are flights that miss the connection with i_4. Hence set $F_{Miss(4)} = \{i_2, i_3\}$. In particular, $i_2 = justMiss(i_4)$ since it has the earlier arrival time amongst the set.

In short, *justMiss(i)* has the following 2 properties in relation to *i*:

1. Arrival station of *justMiss(i)* is the same as the departure station of *i*.
2. *justMiss(i)* has the earliest arrival time among the set of flight legs that would misconnect with *i*.

The identification of *justMiss(i)* for a particular flight *i* is necessary, since it dictates the minimum amount of delay for flight *i* in order to "catch" subsequent flight.

Procedure Initialization:
Sort all existing flight legs by increasing departure time.
Begin:
Pick the first flight leg f_{i,t_i} from the list:
    Determine justMiss(f_{i,t_i}), using the existing batch of flight legs.
    If justMiss(f_{i,t_i}) exists:
        Determine the delay amount of f_{i,t_i} to make a connection to justMiss(i).
            If departTime(f_{i,t_i})+delay amount < recoveryEndTime
            Create f_{i,t_i+1} , a delayed version of f_{i,t_i} by the delay amount to allow connection with justMiss(i).
            Insert newly created f_{i,t_i+1} into current list, in accordance to departure time.
        Remove f_{i,t_i} from the sorted list.
Back to line 4 above

The algorithm above would generate f_{i,t_i+1} , a delayed version of f_{i,t_i} for each f_{i,t_i} investigated. Subsequent iterations would encounter f_{i,t_i+1} eventually and it would in turn generate another delayed version, f_{i,t_i+2} , and so on. Hence the algorithm would continually generate all possible delayed versions of various legs that can make aircraft connections within the time frame of recovery.

As the algorithm iterates, new flight legs are introduced and inserted into the list, in accordance to their departure time, so that at all times the flight legs are sorted in increasing departure time. Each new introduction of flight legs would in turn create opportunities for misconnection for the existing ones currently in the list. The creation of a particular flight leg stops when subsequent delayed version of that flight leg falls outside the recovery end time (line 9). The algorithm terminates when the list is exhausted.

For the purpose of tractability, new versions of flight leg are spaced at least 5 minutes later from its predecessor. In practice, it does not make sense to consider 2 different delayed versions of the same leg in the model that are less than 5 minutes apart.

3.2.4 Binary connection variables

In addition to flight variables, another fundamental variable used in proposed model is binary connection variable.

Binary connection variable x_{i,t_i,j,t_j} has the following physical meaning: it is 1 if the aircraft serving flight f_{i,t_i} serves f_{j,t_j} next. $x_{i,t_i,j,t_j} = 1 \Rightarrow (f_{i,t_i} = 1\, and\ f_{j,t_j} = 1)$. (The implication does not work in the other direction.) For each f_{i,t_i}, a specific x_{i,t_i,j,t_j} is generated for every f_{j,t_j} that can connect from f_{i,t_i}.

The criteria for connection are:

1. Departure station of f_{j,t_j} is same as arrival station of f_{i,t_i}.
2. Departure time of f_{j,t_j} is later or equal to arrival time of f_{i,t_i}, plus a minimum turn time (MTT).

In general the number of connection variables x_{i,t_i,j,t_j} is considerably large, since the combination of possible f_{i,t_i} and f_{j,t_j} is not trivial. However, it turns out the integer constraint on this binary variable can be relaxed in the model. This is in fact the motivation for using this variable in the first place.

There are 3 cases of binary connection variables:

1. Ordinary connection variables connecting flight f_{i,t_i} with f_{j,t_j} (x_{i,t_i,j,t_j}).
2. Connection variables connecting beginning position b with flight f_{j,t_j} ($x_{b,0,j,t_j}$).If $x_{b,0,j,t_j}$=1, it implies aircraft at position b would serve flight f_{j,t_j} next.
3. Connection variables connecting flight f_{i,t_i} with end positions e ($x_{i,t_i,e,0}$).the case of $x_{i,t_i,e,0}$=1, it implies aircraft serving f_{i,t_i} would terminate at position e at end of recovery period.

3.2.5 Passenger itinerary disruption variable

Passengers are modeled in unit of passenger itineraries *p*. A passenger itinerary *p* is defined as groups of passengers following the same series of flight legs, shares the same origin station and intends to reach the same final destination. The entire collection of passenger itineraries p in recovery falls under the set **P**.

Generally a passenger itinerary *p* consists of just 1 single leg. Occasionally, there may be 2 or more connecting flight legs that a particular group of passengers follow. A particular passenger itinerary *p* has the following properties associated with it:

1. One, or a series of connecting flight legs that this itinerary is following. The collection of flight legs in itinerary *p* falls under the set $\mathbf{L_p}$ of *p*.
2. Number of passenger traveling on this itinerary (n_p).

To track the number of passengers disrupted in the recovery model, λ_p, a binary passenger itinerary disruption variable, is introduced. Corresponding to each *p* involved in recovery, a λ_p variable is created.

Passenger itinerary is said to be disrupted when either of the following 2 conditions occur:

1. Any one of the flight legs in set $\mathbf{L_p}$ is cancelled, or
2. Insufficient turn time between a pair of connecting flight legs in $\mathbf{L_p}$. (pairs of connecting flight legs in p falls into set $\mathbf{C_p}$.)

If any of the two conditions above apply, λ_p, passenger itinerary disruption variable, would be true.

3.3 Mathematical Model

$$\text{Min} \quad k_{disrupted} \left\{ \sum_{p \in P} n_p \lambda_p \right\} + \sum_{i,t_i \geq 15 \min Delay} df_{i,t_i} \tag{3.1}$$

Subject to:

$$\sum_{t_i} f_{i,t_i} + f_{i,c} = 1 \qquad \forall i \tag{3.2}$$

$$\sum_{(j,t_j) \in C^+(b,0)} x_{b,0,j,t_j} = 1 \qquad \forall b \in B \tag{3.3}$$

$$\sum_{(i,t_i) \in C^-(e,0)} x_{i,t_i,j,t_j} = 1 \qquad \forall e, e \in E \tag{3.4}$$

$$\sum_{(j,t_j) \in C^+(i,t_i)} x_{i,t_i,j,t_j} = f_{i,t_i} \qquad \forall i, t_i \tag{3.5}$$

$$\sum_{(i,t_i)\in C^-(j,t_j)} x_{i,t_i,j,t_j} = f_{j,t_j} \qquad \forall j, t_j \qquad (3.6)$$

$$\lambda_p \geq f_{i,c} \qquad \forall p \in P, \forall i \in L_p \qquad (3.7)$$

$$\lambda_p \geq f_{i,t_i} + \sum_{\Pi(i,t_i)} f_{j,t_j} - 1 \qquad \forall p, \forall (i,j) \in C_p, \forall (j,t_j) \in \Pi(i,t_i), \forall t_i \qquad (3.8)$$

$$0 \leq x_{i,t_i,j,t_j} \leq 1 \qquad \forall i, \forall t_i, \forall j, \forall t_j \qquad (3.9)$$

$$f_{i,t_i} \in \{0,1\} \qquad \forall i, \forall t_i \qquad (3.10)$$

$$\lambda_p \in \{0,1\} \qquad \forall p \qquad (3.11)$$

Indexes	
i,j	Flight indices
t_i	delay indices for flight i. in particular, t_i=0 denotes flight i's original departure time.
P	passenger itinerary indices

Sets	
B	set of beginning positions of aircrafts at the start of recovery
E	set of end positions of aircrafts at the end of recovery
F	set of all flights involved during recovery

X	set of all connection variables
$C^{+}(j,t_j)$	set of flight variables f_{i,t_i}, whose planes arrives at the same station of departure station of flight f_{j,t_j}, and has sufficient turn time to make the connection from flight f_{i,t_i} to flight f_{j,t_j}
$C^{-}(i,t_i)$	set of flight variables f_{j,t_j}, whose planes departs from the same station of arrival station of flight f_{i,t_i}, and has sufficient turn time to make the connection from flight f_{i,t_i} to flight f_{j,t_j}
P	Set of passenger itinerary involved in recovery
$\mathbf{L_p}$	Set of flight legs involved in passenger itinerary p
$\mathbf{C_p}$	Set of pairs of (i,j) flight connection involved in passenger itinerary p
$\mathbf{\prod(i,t_i)}$	set of flights (j,t_j) that does not manage to connect in time with previous flight (i,t_i), where pair $(i,j)\varepsilon C_p$

Variables	
f_{i,t_i}	Binary variable for flight i at delay index t_i. Is 1 if flight i occurs at time delay t_i. 0 otherwise.
$f_{i,c}$	Is 1 if flight i is cancelled. 0 otherwise
x_{i,t_i,j,t_j}	Binary connection variable. Is 1 if aircraft serving f_{i,t_i} serves f_{j,t_j} next.
λ_p	Binary passenger itinerary disruption variable. Is 1 if passenger itinerary p

	is disrupted, 0 otherwise.

Constants	
$k_{disrupted}$	Costs associated with each passenger disrupted. Expressed in time.
n_p	number of passenger in passenger itinerary p.
d	Costs associated with flight arriving later than 15 minutes after the expected arrival time.

The objective function (3.1) seeks to

1. Minimize the total number of passenger disrupted ($\sum_{p \in P} n_p \lambda_p$)
2. Maximizing the on-time performance of airline.

Given that the equation of on-time performance, as described in chapter 1, mathematically is $=1-\dfrac{(number\ of\ flights\ delayed > 15\,\text{min})}{(total\ number\ of\ scheduled\ flights)}$. After removing the unnecessary constants, we can simplify (2) to become a minimization of $\sum_{i,t_i \geq 15\,\text{min}\, Delay} f_{i,t_i}$.

A cost parameter is attached to passenger disruption ($k_{disrupted}$). At the same time, a cost parameter (d) is attached to flight options that are more than 15 minutes late.

As a result, the objective function becomes

1. minimization of $k_{disrupted}\left\{\sum_{p\in P} n_p \lambda_p\right\}$
2. minimization of $\sum_{i,t_i \geq 15\min Delay} df_{i,t_i}$

Constraint (3.2) enforces the requirement that a particular flight *i* must be either flown at a specific delay option, or be cancelled.

Constraint (3.3) enforces that one, and only one of the connection variables must flow out from a particular starting position *b*.

Constraint (3.4) enforces that one, and only one of the connection variables must flow back into a particular ending position *e*.

Constraint (3.5) enforces that one, and only one of the connection variables flowing from flight f_{i,t_i} must be true, provided f_{i,t_i} is true. Otherwise all of them are false (sum to zero).

Constraint (3.6) enforces that one, and only one of the connection variables flowing out from flight f_{j,t_j} must be true, provided f_{j,t_j} is true. Otherwise all of them are false (sum to zero).

Constraints (3.2) to (3.6) would guarantee us a feasible flight route network for all the aircrafts in recovery.

Constraints (3.7) and (3.8) deal with considerations governing passenger itineraries. Constraint (3.7) dictates that a passenger itinerary disruption variable λ_p will be true if any flight legs in *p* are cancelled.

Constraint (3.8) is used to determine the integer value of λ_p under the non-cancelled flight leg scenario.

Constraint (3.8) is evolved from the following constraint:

$$\lambda_p \geq f_{i,t_i} + f_{j,t_j} - 1 \qquad \forall p, \forall i \in L_p, \forall t_i, \forall (j,t_j) \in \Pi(i,t_i)$$

$\prod(i,t_i)$ contains the set of (j,t_j) that do not manage to connect in time with flight (i,t_i). λ_p will be set to one whenever f_{i,t_i} and any of the subsequent f_{j,t_j} from set $\Pi(i,t_i)$ is true. However, since at any one time at most one f_{j,t_j} can be true, the constraint above can be condensed into:

$$\lambda_p \geq f_{i,t_i} + \sum_{\Pi(i,t_i)} f_{j,t_j} - 1 \;\; \forall p, \forall (i,j) \in C_p, \forall t_i .$$

The revised form above helps to reduce number of constraints in model greatly.

Given the unimodular property of such a network flow setup, connection variables can be relaxed to become continuous. Hence even though constraint (3.9) allow connection variable to vary continuously between 0 and 1, the solution would always yield us an integer solution.

3.4 Conclusion

The chapter describes a basic airline schedule recovery model that, while basic, is interesting in 2 ways. Firstly, it utilizes the concept of flight connection variable. This approach is not commonly encountered in other similar airline recovery research. The unimodular property of the network allows one to relax the flight connection variable into continuous variables. Secondly, the set up of passenger itinerary disruption variable λ_p is such that it can be simplified into a more elegant form, as shown in constraint (3.8). The simplified form involves much less constraints that its original form.

Chapter 4

Extended Model with Maintenance Consideration

The previous chapter focuses on explaining the first mathematical model, which attempt to solve an airline schedule recovery problem through a framework of minimizing number of passenger disrupted and on-time performance. That model is now extended to address the problem of resultant schedule meeting the aircraft scheduled maintenance requirement.

In general airlines perform scheduled maintenance on the aircrafts every few days. The duration and elapsed time between scheduled maintenances differs from airline to airline. These maintenance checks are often provided only at a selected few airports. In general maintenance operation takes up a considerable amount of time, during which aircrafts involved would be grounded for that duration.

4.1 Maintenance Consideration

As noted in the chapter on literature review, aircraft maintenance consideration is not often captured in aircraft recovery models. This is unfortunate since aircraft maintenance occurs regularly and is a major concern for airlines.

There are different types of aircraft maintenances that an airline must fulfill. An aircraft may not be airborne if it has flown for a certain amount of time since the last time it underwent maintenance. The aircraft in question must then undergo maintenance which effectively takes it out of service for a specific duration commonly known as maintenance duration.

There are various types of maintenance needs constraining the usage of aircraft. Generally there is the A check, B check, C check and D check that an aircraft must undergo. The specific duration between maintenance check for these checks actually vary rather significantly from airlines to airlines. In general, the frequency of A check is high enough that one should capture it in the model. The remaining maintenance checks occur too infrequently to be factored into the short range recovery model consideration.

Airlines typically schedule aircrafts to undergo maintenance much earlier before it hits a hard dateline in which flying further would violate flight regulation set down by FAA. By observing this "soft" dateline, airlines have a small buffer to play with to schedule should a disruption occurs.

A further consideration is, not all airports are equipped to perform maintenance check for a particular airline. Generally, only the hub of the airline may perform maintenance check.

Such airports that are equipped to perform maintenance are term maintenance feasible stations.

Hence, the proposed extended recovery model must not only take into consideration the maintenance needs of aircrafts and the duration of maintenance, but also must attempt to fly the aircraft in question to maintenance feasible station.

At the start of recovery time, a subset of aircrafts that require maintenance before the end of recovery time is identified. For convenience these subset of aircrafts are termed maintenance aircrafts. The proposed extended model makes sure the maintenance aircrafts would undergo maintenance before the end of recovery. The model would also make sure that, should the maintenance complete before the end of recovery, the aircraft would be further used in the model. This ensures the model proposes the solution in an optimal way, in the sense that it would not violate maintenance needs of the all the aircrafts while fulfilling its objective of minimizing passenger disruption and maximizing on-time performance.

4.2 Variables and Indexes

Most of the syntaxes adopted over in model 1 are carried over to model 2. However, some slight changes are required to reflect the new aircraft maintenance consideration.

4.2.1 Types of Aircrafts:

To tackle aircraft maintenance needs in a systematic manner, the set of all aircrafts (**A**) involved in recovery operation are divided into 2 types, set $\mathbf{A^1}$ and set $\mathbf{A^2}$. Aircrafts from set $\mathbf{A^1}$ need to undergo maintenance within the duration of recovery. The recovery model must schedule a maintenance slot for aircrafts within this set before the end of recovery. Aircraft from set $\mathbf{A^2}$, however, requires no such special consideration. This difference in maintenance need arises from the recent maintenance history of aircrafts.

For example, aircraft a^1 had its last maintenance check 3 days ago while aircraft a^2 just underwent a scheduled maintenance 2 hours before the disruption. As a result when disruption occurs, aircraft a^1 must undergo maintenance check before the end of recovery period, failure which would result it failing to satisfy the legality governing aircraft maintenance needs. The recovery model must take this into consideration and must schedule a maintenance check for a^1 in its recovered schedule. Hence aircraft a^1 falls under the set $\mathbf{A^1}$. Aircraft a^2, in contrast, does not require special maintenance consideration before the end of recovery period, and hence aircraft a^2 falls under the set $\mathbf{A^2}$. Recovered schedule does not need to pay special maintenance attention to aircrafts falling under set $\mathbf{A^2}$.

4.2.2 Types of Beginning and Ending positions:

Due to this classification, start nodes are divided into 2 types, namely start node type 1 (b^1, collectively forming the set $\mathbf{B^1}$) and type 2 (b^2, collectively forming the set $\mathbf{B^2}$). b^1

indicate positions of aircrafts a^1 at the start of recovery, while b^2 indicate position of aircrafts a^2 at the start of recovery.

End nodes (e) are not classified likewise. It is sufficient that all aircrafts terminate their flight path at any of these end nodes.

4.2.3 Types of Connection Variables:

To make distinction between aircrafts that still require maintenance consideration from aircrafts that do not, there are now 2 sets of connection variables, connection variables type 1 (x^1) and type 2 (x^2). Recovery model would traverse out a flight path for aircraft type 2 using only x^2. Flight path for aircraft type 1, on the other hand, would traverse a flight path using x^1 initially. Once it meets a maintenance slot it would then continue traversing using x^2 until it terminates at the end. This is further explained in the paragraphs below.

x^1 are further divided into 2 types, maintenance connection variables and non-maintenance connection variables. The former, identified as $x^{1,m}$, falls into the set **M.** The latter, $x^{1,\bar{m}}$, falls under the set $\overline{M}$. For a type 2 connection variable to qualify in set **M**, the arrival station of incoming flight i (or the departure station of outgoing flight j) must be a maintenance station. In addition, the elapsed time in between arrival time of incoming flight and departure time of outgoing flight must be greater than or equal to the minimum required maintenance time.

The two paragraphs above are best explained by illustrating differences of the path traversed by type 1 and 2 aircrafts in the final solution. (To describe a path traversed by a particular aircraft in the final solution, it is sufficient to identify the beginning position *b*, the series of connection variables linking the path, and the ending position *e*.)

b^2 connect with connection variables from set X^2 and each forms a route that terminates at any of the end points *e*. The resultant route describes the path to be traversed by the aircraft a^2.

Type 1 start nodes b^1 connect with connection variables from set $\mathbf{X^1}$, and in between b^1 and any end points *e*, the path must traverse through a maintenance connection variable $x^{1,m}$, after which it connects with connection variables from set $\mathbf{X^2}$ and terminate at any end point *e*.

The requirement that aircrafts a^1 must go through $x^{1,m}$ ensures these aircrafts will actually have opportunity to undergo maintenance, thus fulfilling their maintenance needs. This requirement is enforced by not creating any x^1 that terminates at any end nodes *e*. Instead "crossing over" between $x^{1,m}$ and x^2 is allowed. Hence, the model forces aircrafts a^1 to find a path to the end points via any $x^{1,m}$, crossing over to x^2, and finally terminate at end point *e*. The only exception to this requirement is when maintenance connection variable link directly to end point *e*.

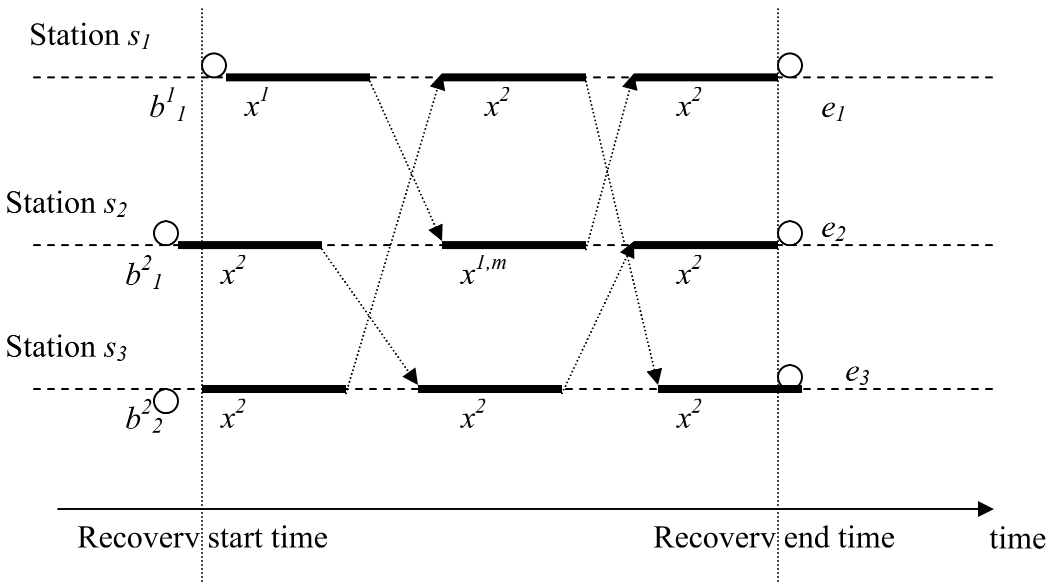

Figure 4.1: Possible Flight Connections

Using the scenario above, 3 aircrafts are involved in recovery. In particular, aircraft at position $b^1{}_1$ requires maintenance before the end of recovery period. However, out of the 3 stations, only station s_2 provides maintenance service. Aircraft first traverse using connection variable type 1, and eventually terminates at end point e_1, through a connection variable type 2, made possible via a maintenance connection variable $x^{1,m}$. In the other 2 cases, aircraft from position b^2 traverse flight path described solely using connection variables type 2.

4.3 Mathematical Model

$$\text{Min} \quad k_{disrupted}\left\{\sum_{p\in P} n_p \lambda_p\right\} + \sum_{i,t_i \geq 15 \min Delay} df_{i,t_i} \tag{4.1}$$

Subject to:

$$\sum_{t_i} f_{i,t_i} + f_{i,c} = 1 \qquad \forall i \tag{4.2}$$

$$\sum_{(j,t_j)\in C^+(b,0)} x^1_{b,0,j,t_j} = 1 \qquad \forall b \in B_1 \tag{4.3}$$

$$\sum_{(j,t_j)\in C^+(b,0)} x^2_{b,0,j,t_j} = 1 \qquad \forall b \in B_2 \tag{4.4}$$

$$\sum_{(i,t_i)\in C^-(e,0)} x^{1,m}_{i,t_i,j,t_j} + \sum_{(i,t_i)\in C^-(e,0)} x^2_{i,t_i,j,t_j} = 1 \qquad \forall e, e \in E \tag{4.5}$$

$$\sum_{(j,t_j)\in C^+(i,t_i)} x^1_{i,t_i,j,t_j} + \sum_{(j,t_j)\in C^+(i,t_i)} x^2_{i,t_i,j,t_j} = f_{i,t_i} \qquad \forall i, t_i \tag{4.6}$$

$$\sum_{(i,t_i)\in C^-(j,t_j)} x^1_{i,t_i,j,t_j} + \sum_{(i,t_i)\in C^-(j,t_j)} x^2_{i,t_i,j,t_j} = f_{j,t_j} \qquad \forall j, t_j \tag{4.7}$$

$$\sum_{i,t_i,t_j} x^{1,\overline{m}}_{i,t_i,j,t_j} \leq \sum_{t_j,k,t_k} x^{1}_{j,t_j,k,t_k} \qquad \forall j \in F \qquad (4.8)$$

$$\sum_{i,t_i,t_j} x^{1,m}_{i,t_i,j,t_j} + \sum_{i,t_i,t_j} x^{2}_{i,t_i,j,t_j} \leq \sum_{t_j,k,t_k} x^{2}_{j,t_j,k,t_k} \qquad \forall j \in F \qquad (4.9)$$

$$\sum_{i,t_i,t_j} x^{1}_{i,t_i,j,t_j} = n_j \qquad \forall j \in F \qquad (4.10)$$

$$\lambda_p \geq f_{i,c} \qquad \forall p \in P, \forall i \in L_p \qquad (4.11)$$

$$\lambda_p \geq f_{i,t_i} + \sum_{\Pi(i,t_i)} f_{j,t_j} - 1 \qquad \forall p, \forall (i,j) \in C_p, \forall n \qquad (4.12)$$

$$0 \leq x^{1}_{i,t_i,j,t_j} \leq 1 \qquad \forall x^{1}_{i,t_i,j,t_j} \in X^1 \qquad (4.13)$$

$$0 \leq x^{2}_{i,t_i,j,t_j} \leq 1 \qquad \forall x^{2}_{i,t_i,j,t_j} \in X^2 \qquad (4.14)$$

$$n_j \in \{0,1\} \qquad \forall j \in F \qquad (4.15)$$

$$f_{i,t_i} \in \{0,1\} \qquad \forall i, \forall t_i \qquad (4.16)$$

$$\lambda_p \in \{0,1\} \qquad \forall p \qquad (4.17)$$

Indexes	
i,j	Flight indices
t_i	delay indices for flight i. in particular, t_i=0 denotes flight i's original

	departure time.
P	passenger itinerary indices

Sets and Variables	
Aircrafts:	
A	set of all aircrafts
$\mathbf{A^1}$	set of aircrafts requiring maintenance before the end of recovery period. Subset of **A**
a^1	aircraft index of $\mathbf{A^1}$
$\mathbf{A^2}$	Complement set for $\mathbf{A^1}$ that completes **A**
a^2	aircraft index of aircrafts in set $\mathbf{A^2}$
Positions:	
B	set of beginning position of planes at start of recovery
$\mathbf{B^1}$	set of beginning position of aircrafts from set $\mathbf{A^1}$
b^1	Index of beginning position of $\mathbf{B^1}$
$\mathbf{B^2}$	set of beginning position of aircrafts from set $\mathbf{A^2}$
b^2	Index of beginning position of $\mathbf{B^2}$
E	set of end position of planes at end of recovery
e	Index of end position of **E**
Flight and Connection Variables:	

F	Set of all flights involved during recovery period.
f	An index of **F**
f_{i,t_i}	binary variable flight i at delay index t_i
$f_{i,c}$	=1 if flight i is cancelled. 0 otherwise
t_i	delay index for flight i
X	Set of connection variables (cv)
x^1_{i,t_i,j,t_j}	• Connection variable for aircraft from set A_1. If plane a_1 of flight i (coming in at delay time t_i) connects to flight j (leaving at delay time t_j), then x^1_{i,t_i,j,t_j} =1. otherwise 0. • The route that chains the various x^1_{i,t_i,j,t_j} emerges from the start nodes of set B_1. (see model later) • None of the cv of set X_1 may terminate at any end nodes E.
x^2_{i,t_i,j,t_j}	• Connection variable for aircraft from set A_2. If plane a_2 of flight i (coming in at delay time t_i) connects to flight j (leaving at delay time t_j), then $x_2(i,t_i,j,t_j)$=1. otherwise 0. • The route that chains the various x^2_{i,t_i,j,t_j} emerges from the start nodes of aircrafts set A^2. (see model later). • The route that chains the various x^2_{i,t_i,j,t_j} terminates at end nodes.
$\mathbf{X}^1$	Set of connection variables of x^1

X^2	Set of connection variables of x^2
$C^+(j,t_j)$	set of flight variables f_{i,t_i}, whose planes arrives at the same station of departure station of flight f_{j,t_j}, and has sufficient turn time to make the connection from flight f_{i,t_i} to flight f_{j,t_j}.
$C^-(i,t_i)$	set of flight variables f_{j,t_j}, whose planes departs from the same station of arrival station of flight f_{i,t_i}, and has sufficient turn time to make the connection from flight f_{i,t_i} to flight f_{j,t_j}.
M	Set of combinations of (i,t_i,j,t_j) that have the following properties: • (i,t_i,j,t_j) forms a valid connection variable. • The arrival/departure station of incoming flight $f(i,t_i)$/outgoing flights $f(j,t_j)$ is maintenance station (stations that offer maintenances) • The difference between the arrival time of incoming flight (t_i+$BlkTime_i$) and departure time of outgoing flight (t_j) is greater than the required time for maintenance.

$\overline{M}$	Set of combinations of (i,t_i,j,t_j) that :
	
	• (i,t_i,j,t_j) forms a valid connection variable, and
	    o The arrival/departure station of incoming flight f_{i,t_i} /outgoing flights f_{j,t_j} is not maintenance station (stations that offer maintenances); or
	    o The difference between the arrival time of incoming flight

	$(t_i+BlkTime_i)$ and departure time of outgoing flight (t_j) is less than the required time for maintenance
Passengers:	
P	set of passenger itinerary
p	passenger itinerary index
L_p	set of flight legs involved in passenger itinerary p
λ_p	binary variable for disrupted itinerary p. Equals 1 if itinerary p is disrupted, 0 otherwise
C_p	set of pairs of (i,j) flight connection involved in passenger itinerary p
$\prod(i,t_i)$	set of flights (j,t_j) that does not manage to connect in time with previous flight (i,t_i), where pair $(i,j) \varepsilon C_p$

Constants	
d	Delay penalty incurred for delays more than 15 minutes.
n_p	number of passengers on itinerary p
$k_{disrupted}$	penalty delay costs incurred per- disrupted passenger

The integrality constraint of connection variables from set X (equation 12 and 13) has been relaxed to continuous variables.

The objective function of model (4.1) remains the same: To minimize number of passenger disrupted/inconvenienced, while at the same time attempt to preserve on time performance of airline.

Constraint (4.2) dictates that a particular flight f_i either gets flown at a particular delayed time, or get cancelled.

Constraint (4.3) dictates that only one of the various connection variables x^1 emerging from start node b^1 is chosen.

Constraint (4.4) dictates that only one of the various connection variables x^2 emerging from start node b^2 is chosen.

Constraint (4.5) dictates that only connection variables from set X^2 may terminate at end point e. Exceptions are maintenance connection variables from set $X^{1,m}$ that happens to terminate at end point e too.

Constraint (4.6) serves the same purpose as before: to ensure all the connection variables with incoming flight i actually originates from flight i.

Constraint (4.7) ensures all the connection variables with outgoing flight j actually connect to flight j.

Constraints (4.8) and (4.9) are termed as sequencing constraints. Constraint (4.8) ensures that, whenever a predecessor x^1 is true, then one of the potential successors x^1 must be true too. This has the effect of ensuring the sequence of connection variables is faithfully kept within the same type. However, connection variables from set $X^{1,m}$ are exempted from observing this rule and is taken care of in constraint (4.9).

Constraint (4.9) ensures that, if predecessor is a $x^{1,m}$ or x^2, then the successor must be a x^2. Constraint (4.9) allows the "switching over" described in earlier section. A flight path is allowed to go from x^2 to x^3 via $x^{2,m}$ via this equation.

Equation (4.10) is necessary to avoid situations of splitting of values. With the introduction of 2 types of connection variables, the uni-modular structure of the previous model is destroyed. This is because for a particular combination of (i,t_i,j,t_j), there can be 2 types of connection variables competing for flow. Relaxing the integer requirement of connection variables frequently results in non-integer solution, where flow is split evenly between x^1 and x^2. Equation (4.10) ensures that the sum of all connection variables x^1 outgoing via the same flight j, must equal to an integer solution. This will "tie" the end of connection variables sharing the same outgoing flight j and will be unable to "share" the flow with type 2 variables. The introduction of (4.10) ensures integral solution in the final solution, despite the relaxed constraint.

Equations (4.11) and (4.12) are the same as before: Equation (4.11) ensures that all the passengers flying originally in flight i would be registered as disrupted whenever $f_{i,c}$ is true. Equation (11) registers disrupted passengers whenever there is insufficient connect time for passenger in between incoming and outgoing flight.

Finally, the integrality constraint on connection variables (12) and (13) can be relaxed.

4.4 Conclusion

The expanded model from the model described in chapter 3 is also novel in a few manners. It is a fresh attempt to handle flight maintenance needs of aircrafts through the use of connection variables. Conceptually, there is nothing stopping one from expanding it to more than two types of connection variables to meet finer needs of maintenance requirements of aircrafts. As a result, the model is also expandable in this sense.

Chapter 5

SimAir : Simulation of Airline Operation

The mathematical models proposed in previous two chapters are tested against a moderately sized real life flight schedule. To obtain meaningful measure of average number of passenger disrupted and on-time performance, an airline schedule operation simulation software is used. This chapter is devoted to describing the airline schedule operation simulation software, which is part of the work done during the course of thesis.

5.1 SimAir Background Information

In order to prove the validity and practicality of the proposed model, the recovery mathematical model is incorporated into SimAir. SimAir is an airline operation simulation software developed jointly between National University of Singapore and Georgia Institute of Technology.

SimAir is a discrete event simulation of airline operations. Its purpose is two-fold. It is meant to be a research tool to help in evaluating effectiveness of a particular airline schedule recovery policy. One may also use it to evaluate the robustness of a given schedule in operations. These evaluations are estimated when user uses SimAir to simulate the various unexpected disruptions during an airline operation.

SimAir can simulate the flight operations of aircrafts, crews, passengers and cargo flow. Depending on input, SimAir may simulate out the entire weekly schedule of an airline, tracking the movement of related crews, passengers and cargoes through the flight network.

5.2 SimAir Conceptual Model

SimAir is made up of three main modules namely simulation module, controller module and recovery module. The organization of the conceptual modules is shown in figure below.

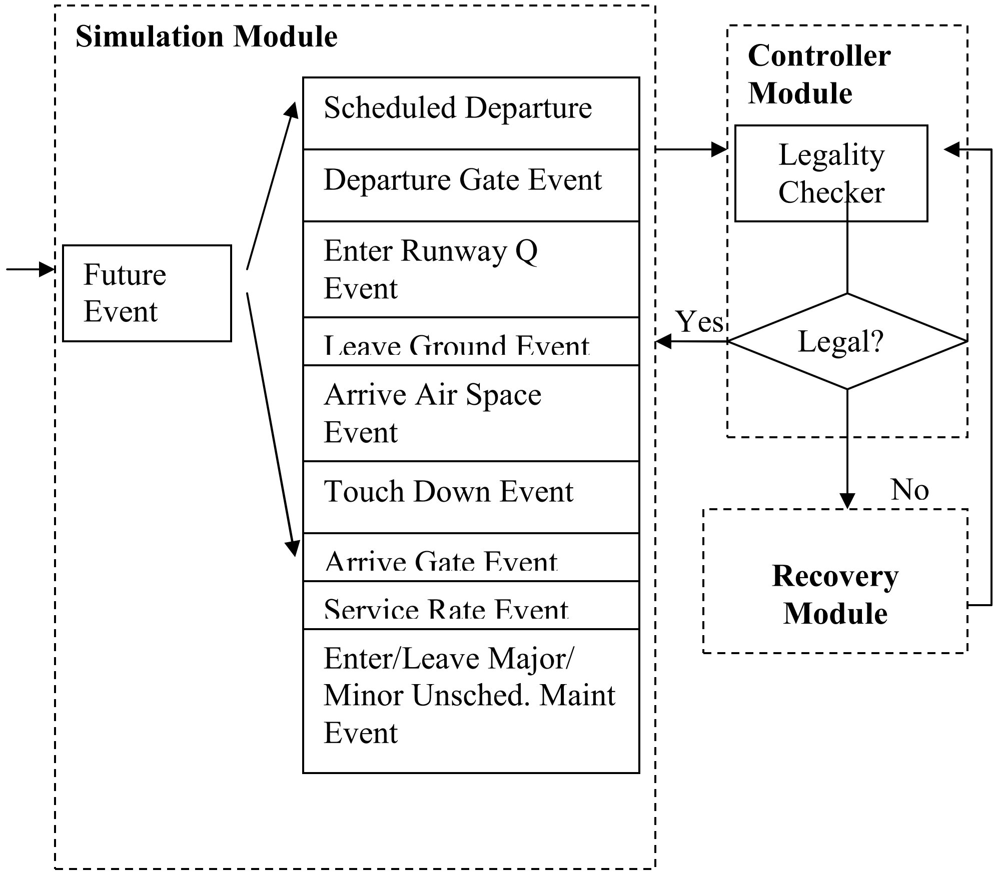

Figure 5.1: Modular Structure of SIMAIR

A flight schedule is made up of flight legs that the airline will fly with its fleet, crew and the involved passenger itineraries. The schedule is read into the simulation module in a pre-defined format. SimAir then simulates the operation using the schedule provided. The simulation duration of airline operations in SimAir is dependent on the input schedule provided. The user can opt to run the simulation over many replications, to obtain aggregate results.

5.3 Simulation Module

The simulation module is solely involved in simulation operations. It is made up of a few components. There is a future event list, an event scheduler, and a simulation clock.

The simulation module models the aircraft's operation as a sequence of events. One event triggers another leading to a series of airline operations.

Each flight leg in the schedule can be decomposed down to seven events.

i. Scheduled departure event – pilot and passenger scheduled to depart from the gate.
ii. Departure gate event - aircraft pushes away from the gate and begins to taxi to the runway.
iii. Enter runway queue event – aircraft enters the runway queue of the departure station.
iv. Leave ground event – aircraft reaches the front of the runway queue and begins its flight.
v. Arrive airspace event – aircraft enters the airspace queue of the arrival station.
vi. Touch down event – aircraft reaches the front of the airspace queue and begins to land.
vii. Arrive gate event – aircraft reaches the gate at the arrival station.

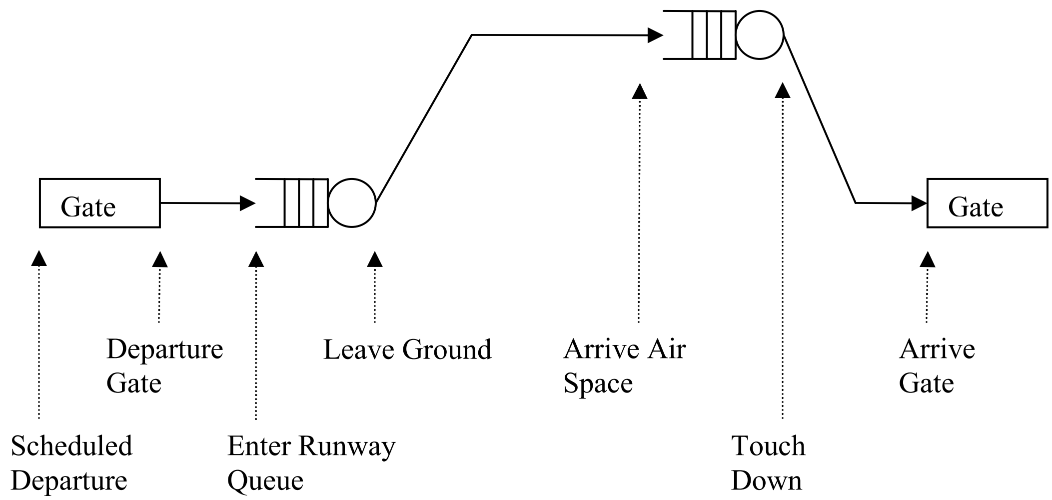

Figure 5.2: Decomposition of a Flight Leg

In addition to the seven events which decompose a leg above, there are an additional five events

i. Enter major unscheduled maintenance event – aircraft is required to undergo major unscheduled maintenance. This is a chance event that is generated after departure gate event.
ii. Enter minor unscheduled maintenance event– same as above, but is a minor unscheduled maintenance, and the maintenance duration is shorter.
iii. Leave major unscheduled maintenance event– complement of (i), it is generated after aircraft goes under unscheduled major maintenance, and signals the simulation module that the aircraft is now ready to fly again.
iv. Leave minor unscheduled maintenance event – similar to (iii), it is the complement of (ii).

v. Service rate event – an event that changes the service rate of runway of airports. This event changes the duration of aircraft taking off and landing. When service rate drops to zero, the airport is closed and no aircrafts can take off or land. The current version of SimAirdoes not explicitly model the effect of other airlines or weather. Such effects are reflected as a change in service rate of the airport.

The SimAir model describes the operation of a particular airline or a particular fleet of an airline. The effect that other airlines and weather have on the congestion of an airport is modeled as the service rate of the airport. To illustrate, under normal conditions the service rate of an airport would be high, where more aircrafts can land/takeoff. Under conditions considered worse than normal, the rate would be lower.

At each station, aircrafts are modeled to fly-in and fly-out as a first-in-first-out queue. To simulate this action, a runway queue and airspace queue are modeled. The queues are assumed to have an infinite capacity. How fast these aircrafts take off or land, depends on the service rate of the station. The service rate of the station in turn is a function of weather condition. The service rate is also used to describe the congestion condition at a station (due to aircrafts of other airlines) indirectly.

Passengers are simulated at the level of itineraries in SimAir. An itinerary represents a group of passengers flying the same set of flights, starting from a departure station, to the intended end destination. During the course of simulation an itinerary may be split into 2 or more itineraries due to disruptions.

The various durations between events follows some distributions that is user configurable. In addition, the probabilities of unscheduled maintenance events are also distributions that are user configurable. These stochastic distributions are based on distributions collected from real life scenarios. Collectively, they make the entire simulation highly stochastic.

5.4 Controller Module

The simulation module, in the course of execution, calls the controller module at the beginning of every event. The controller module checks for rules and regulations that govern the duration of operational hours of aircrafts and crews. Such regulations are usually mandatory rules enforced by Federal Aviation Administration (FAA).

Each event in the simulation is associated with a corresponding controller in SimAir. On occurrence of an event, the corresponding controller checks for the legality of the current schedule. Legality rules based on those obtained from a major US carrier. They can be broadly classified as pertaining to Plane legality and Crew Legality.

Maintenance issues like *max-day* rule, *max-cycle* rule, and *max-block-time* rule, are taken care of while ensuring Aircraft legality. Generally, aircraft must undergo plane maintenance within the limit specified by these three rules. So, at specified events, the plane is checked to see if the *max-day* or *max-fly* time has been exceeded. An aircraft requires different types of maintenances and the duration between each maintenance type is limited. Aircraft are not allowed to take off if they have not performed a particular maintenance type for a period exceeding the said duration above. *Max-block-time* rule limits the accumulated block time an aircraft can fly before it must perform a maintenance. *Max-day* rule limits the duration after which an aircraft performed its last maintenance and before it has the next scheduled maintenance. Finally *max-cycle* rule limits the number of times the aircraft can take off before the next scheduled maintenance. Depending on the flight schedule, any one of these three factors can be the limiting factor forcing the aircraft to stop flying and perform its obligatory maintenance check.

At the start of simulation, the user can input the maintenance history for each of the planes i.e. the number of days, hours and the cycles since the last maintenance. For rule checking involving maintenance, this history would be used to initialize the time at which maintenance was last done for each plane.

For crews, currently three crew legality rules are checked in SIMAIR (although the modular structure of the code allows additional legality rules to be coded in with relative ease.) The three crew legality rules are maximum duty rule, *8-in-24-hour* rule and *30-hours-in-7-days* rule.

The *max-duty* rule ensures that the duty time contained in a single duty is always bounded. The upper bound on the maximum duty time allowed in a single duty is decided depending on the start time of the duty as shown in Table 1.

Duty Start Time	Max allowed duty hours
0500-1659 hours	14.00 hours
0700-1959 hours	13.00 hours
2000-0459 hours	12.00 hours

Table 5.0: Interpretation of max-duty rule

The users have to note that the duty usually starts with a briefing time and ends with debriefing time.

The *30-hours-in-7-days* rule ensures that in any given period of 7days, a crew should not fly more than 30 hours.

The *8-in-24-hour* rule has to be interpreted from the Table 2 given below.

Scheduled Aloft in 24 hours	Hours of Scheduled Rest	Hours of reduced rest	Hours of compensatory rest
<8 hours	≥9.5	8-9.5	10
8<blockTime<9	≥10	8-10	11
>9hrs	≥11	9-11	12

Table 5.1: Interpretation of 8-in-24-hour rule

This rule ensures that the crew can not take off if the last rest within the 24hour look-back from current point of time is less than 8 hours. There must be at least 8-hour rest in the look back period of 24 hours. However depending on the block-time in the 24-hour look back as shown in Table 2, a certain reduced rest shorter than 8 hours is allowed, provided there is compensatory rest provided at the end of the duty. The amount of compensatory rest given is determined by the scheduled hours aloft and the actual hours of reduced rest, as shown in Table 2.

Apart from these aircraft and crew legality rules a set of Common Checks are performed at SDE like availability of plane, crew and also the service rate of destination airport (to confirm if the destination airport is closed).

Users of SIMAIR may code and add their own rules or customize the rules to be checked at every event. If the controller detects infeasibility in the current plan due to a disruption, it calls recovery. After the recovery module suggests a solution, the controller checks the immediate feasibility of the proposed new plan and implements the proposed solution if the next immediate leg is feasible.

The users can note that, the rule checking at each event can be disabled as per users' choice. More on this rule configuration input file can be found in the chapter *Input Files*.

5.5 Recovery Module

A general framework for the recovery module has been established. Currently, a default recovery policy is in place. The default recovery policy utilizes a set of simple heuristics to recover from the disruptions, and is mostly concerned with resolving immediate illegalities. The set of recovery actions used are:

- Pushback of flights when the delay is lower than a threshold and still maintains schedule feasibility.
- Short cycle cancellation of flights in the event that pushback is infeasible.
- Diverting aircraft in the air to alternative airports when destination airport is closed, or aircraft are about to run out of fuel.
- Putting legs “on hold” when a major disruption occurs, such as airport closed down. Flights are prevented from continuing, and only released from on hold status when situation recovers (airport reopen).
- Ferrying of aircraft to stations with maintenance capability to ensure maintenance feasibility.

The modular structure of SimAir makes it a logical and convenient choice to validate the usefulness of the two mathematical models proposed. The subsequent chapters show the simulation results obtained by incorporating the two mathematical models proposed. The results are then compared against the default recovery procedure currently employed by SimAir utilizing a set of heuristics, as a gauge to measure the effectiveness of the two models proposed.

5.6 Conclusion

There is a need to validate the two mathematical recovery models proposed in chapter 3 and chapter 4 respectively. To this end, an airline schedule simulation software is coded and used. SimAir is designed from ground up in a modular manner. The idea is to allow researchers to plug in portion of codes as needed. In this instance, the idea is to utilize SimAir to test out the effectiveness of recovery models proposed.

The following chapter describes the experimental setup of SimAir to meet the need of the objective above.

Chapter 6

Airline Schedule Recovery Results using SimAir

SimAir is used to test out the effectiveness of the 2 recovery models proposed in this work. The default recovery model which comes with SimAir, that utilizes heuristics rules, is used to act as a basis for comparison.

A set of SimAir simulation is performed using default recovery model. A second set of SimAir simulation is performed using mathematical model proposed in chapter 4. A comparison is then done on the results of simulation for these 2 recovery models.

The simulation settings used, the results obtained as well as comparison of simulation results, are described in this chapter.

6.1 Approach to Handle Multi-Objective problem

The direction of this thesis taken is to tackle the two opposing objective of the airline problem: To minimize number of passenger disrupted/inconvenienced, while at the same time attempt to preserve on time performance of airline.

The approach to tackle the multi-objective nature of this problem is to first generate mathematical models with a single objective functions composed of 2 objective terms. This is done in chapter 3 and 4 respectively. In addition, a simulation program is written to generate a large variety of flight scenarios for the models to be tested against. The weights of the two objective terms are varied, and the generated scenarios are solved, to determine how well the solution performed, against the dictated weights. In the end, a set of pareto-optimal solution fronts are generated against the 2 varied objective weights. All the solutions are equally good on the optimal front, and it is up to the respective airline company to decide which of the solution they wish to pick, depending on their company's policy.

6.2 Simulation Settings

6.2.1 Airline Legality Rules used

SimAir is capable of simulating flight, aircraft maintenance, crew and passenger legalities. Considering that the said mathematical model in chapter 4 does not resolve flight maintenance check and crew maintenance checks, only passenger legality checks and flight legality checks are performed for the entirety of the 2 sets of simulation.

6.2.2 Schedule used for Simulation

For the purpose of this report, a flight schedule utilized by a major airline in United States for a single fleet type is used as schedule input. This moderately sized schedule involves 82 aircrafts, 2464 scheduled flight legs, 45 stations and spans over 7 days. On average, there are up to 352 flight legs each day. Disruptions are created to the schedule during the simulation based on various duration delay data inputted. In addition, there are a total of 7944 passenger itineraries, involving 95975 individual passengers. About 69% of the passenger itineraries involve more than 2 connecting flight legs, with inter-flight turn time of less than 1 hour apart. Regular flight maintenance checks are turned off during the duration of simulation.

6.2.3 Settings for Objective Function

The objective of the model proposed is chapter 4 is:

Minimize $k_{disrupted}\left\{\sum_{p\in P} n_p \lambda_p\right\} + \sum_{i,t_i \geq 15\min Delay} df_{i,t_i}$

The two objectives above (individual passenger delay cost, $k_{disrupted}$ and flight late arrival cost, d) are varied to study the effects of variation upon the results of simulation, there by attempting to find a good pareto front that yields the optimal combination of values to provide good recovery solutions.

The two objectives above are varied in the following manner:

Flight delay cost, *d*:

- 1000, 2000, 5000, 10000, 20000, 40000, 50000

Individual passenger disrupted cost, $k_{disrupted}$:

- 300 (5 hours), 360 (6 hours).

The rationale for setting $k_{disrupted}$ at 5 hours and 6 hours each is based on estimate that, for each passenger that gets disrupted, on average they would be delayed from reaching their final destination up to 5 to 6 hours long. Flight delay cost, *d*, on the other hand, is varied from an arbitrary small value of 1000 up to a large value of 40000. There is no factual basis for the selection of this range since it very much depends on the emphasis of individual airlines to the importance of on-time performance statistics.

A total of 14 settings (7*2) are generated based on combination above. Each setting is run for 20 iterations each, to allow a good aggregate of values. Each iteration of simulation is run for one week's worth of schedule mentioned above. During the simulation of the one week's schedule, various delays and unexpected disruptions such as unscheduled maintenance are created. In other words, each value in the cells reported below represents aggregate of values of over 20 iterations of simulations, with multiple recovery procedures occurring within each iteration of simulation itself.

On average, there are about 12 significant disruptions that require SimAir to call for recovery module to solve. This translates into an average of 1.7 disruptions per-day, a close imitation to real world scenario.

6.2.4 Simulation Results to Collect

At the end of each iteration, SimAir would consolidate the whole week's worth of schedule's performance and output a summary of the airline's performance. Hence in total 280 simulations are run, and 14 sets of consolidated data are obtained. For the purpose of discussion the following data are collected at the end of simulation run:

- Number of passenger disrupted
- On-time percentage performance of airline for the 1 week's schedule
- Number of flights cancelled.

More formally, a passenger is said to be disrupted if one of the following 2 scenarios occurred:

1. One or more of the flight legs in its itinerary is cancelled.
2. Delay of an earlier flight leg in its flight itinerary causes it to have insufficient turn time to catch the following flight in its itinerary.

On-time percentage is calculated as:

$$\frac{number\ of\ \ flights\ arriving\ less\ than\ 15\min\ later\ than\ scheduled\ arrival\ time}{number\ of\ \ flights\ actually\ \ flown\ in\ the\ 1\ week\ schedule}\times 100$$

6.2.5 Hardware and Software Specification

The desktop used to run the simulation has an Intel Pentium III 3.0GHz processor. The computer also has 1 G RAM memory. To remove biases, all 3 recovery model simulations are run on the same desktop.

To perform optimization for the two proposed mathematical models, ILOG Cplex 3.7 is used. In general there is no tweaking used beyond the set of default optimization rules utilized within ILOG Cplex.

6.3 Simulation Results

For purpose of comparison, the default recovery model utilized by SimAir, is also run and results are consolidated. The default recovery model in SimAir utilizes a simple set of heuristic rules, with very local views, to help it to perform recovery. In general, if a flight is less than 30 minutes delayed, push-back is used. Short cycle cancellation is considered whenever there is more than 30 minutes delay. Aircraft swapping is not allowed in the default recovery.

6.3.1 Simulation Results ran using SimAir Default Recovery

The following simulation results are average results, with SimAir default recovery, collected over 20 iterations of SimAir simulation run.

On average, a total of 9823.66 passengers (approximately 10.23% of total passengers simulated) are disrupted at the end the week. It has an on-time performance index of 85.88%, and an average of 28 flights is cancelled each week.

Since flight maintenance legality check is disabled within SimAir, flight maintenance legality check is ignored during simulation. On average, 2.4 flight maintenance on average a total of 2.4 maintenance violations occur in a week's worth of schedule.

6.3.2 Simulation Results ran using Recovery Model proposed

SimAir is run using the exactly same setup, schedule and random number seeds to run for the 10 scenarios mentioned above. The simulation results are tabled below.

On-time performance		
	k_disrupted	
delay cost *d*	300	360
1000	86.0	85.0
2000	86.5	85.3
5000	87.1	85.6
10000	88.1	87.1
20000	88.5	88.1
40000	89.7	88.7
50000	89.9	89.0

Table 6.1: On-Time Performance Results using Proposed Recovery Model

Passenger Disrupted				
delay cost *d*	*k_disrupted*			
	300		360	
	# Pax Disrupted	% Pax Disrupted	# Pax Disrupted	% Pax Disrupted
1000	4006.96	4.175	3980.08	4.147
2000	4543.13	4.733	4389.69	4.573
5000	4758.44	4.958	4525.22	4.715
10000	5002.22	5.212	4978.22	5.187
20000	5229.68	5.449	5102.99	5.317
40000	5332.37	5.556	5179.77	5.397
50000	5399.13	5.626	5284.27	5.506

Table 6.2: % Passenger Disrupted Results using Proposed Recovery Model

Number of Flights Cancelled		
delay cost *d*	*k_disrupted*	
	300	360
1000	12.7	9.2
2000	13.1	10.3
5000	14.7	10.8
10000	20.0	14.7
20000	32.0	31.0
40000	29.3	33.2
50000	30.1	34.5

Table 6.3: Number of Cancelled Flights Results using Proposed Recovery Model

Each cell represents an average score of over 20 simulation results ran using the same scenario.

On average, there are 12 disruptions occurring over the one week's worth of schedule. Each disruption would cause simulation to be paused and SimAir would invoke the recovery model to resolve the problem. Each disruption and subsequent recovery would involve 350 flight legs, and take up to 25 seconds to resolve each instance of disruption. In addition, since flight maintenance is not considered, there are occasions where maintenance needs are violated. However, there is no noticeable trend in the maintenance violation: on average a total of 2.4 maintenance violations occur in a week's worth of schedule.

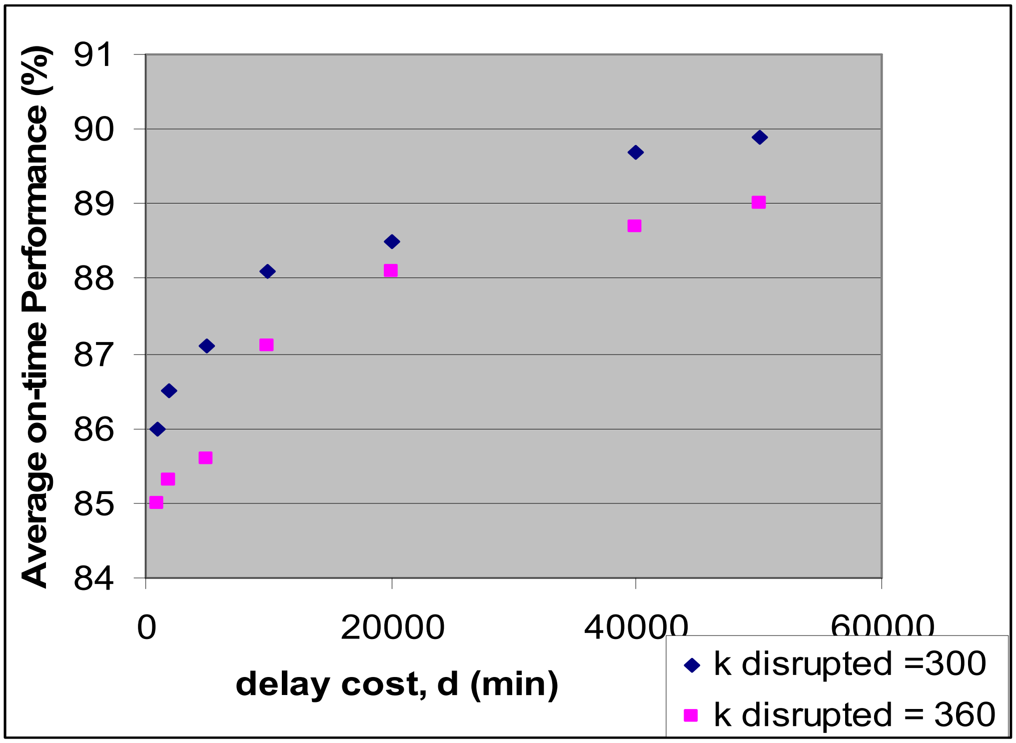

Figure 6.1: On-time Performance vs Delay Cost *d*, using Proposed Recovery Model

In general as delay cost *d* increases, the on-time performance improves. The more emphasis is given to the delay cost, the more the model attempt to improve on the on-time performance.

In addition, given the same delay cost *d*, the on-time performance of the setting with lesser $k_{disrupted}$ is better. This again agrees with intuition since, all else being equal, a lesser emphasis on $k_{disrupted}$ allows the model more leeway to find a schedule that departs earlier.

The improvement in on-time performance index with increasing delay cost d would come at the price of more flight legs being cancelled and passengers getting disrupted.

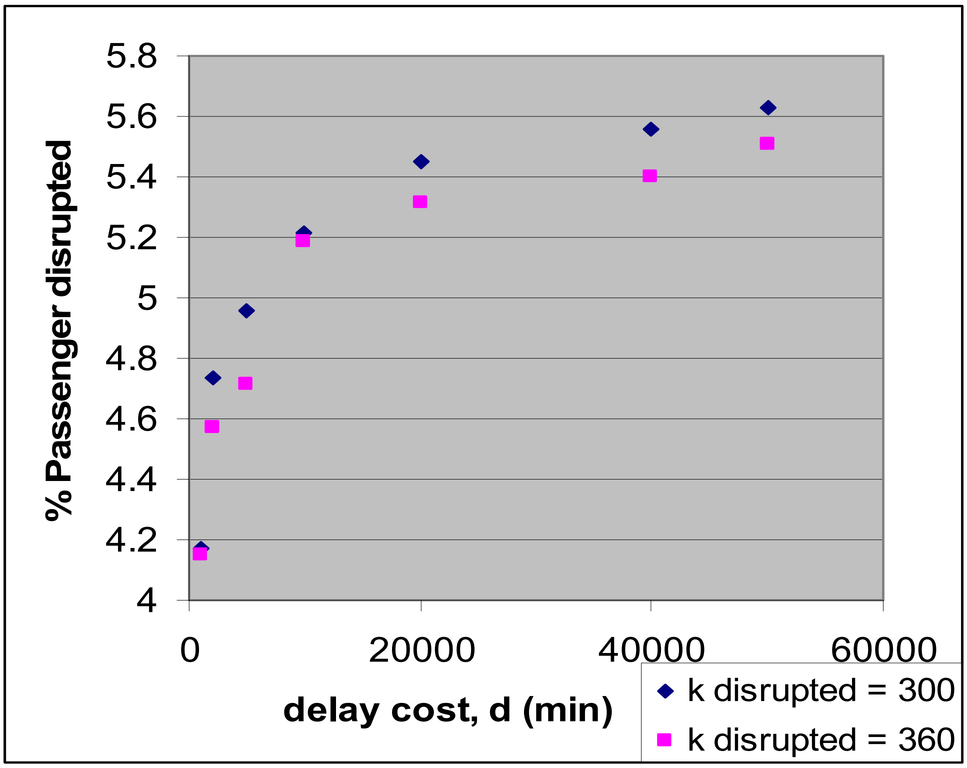

Figure 6.2: % of Passenger Disrupted vs Delay Cost d, using Proposed Recovery Model

The chart above shows the number of passenger disrupted against delay cost d. Two plots are given, with the former with setting $k_{disrupted}$ set at 300, and the latter setting $k_{disrupted}$ at 360.

Once again, with increasing delay cost d, more passengers are getting disrupted from their original itinerary. This agrees with what has been concluded above. Across the two plots, all else being equal, a setting with $k_{disrupted}$ at 360 perform consistently worse than a setting with $k_{disrupted}$ at 300.

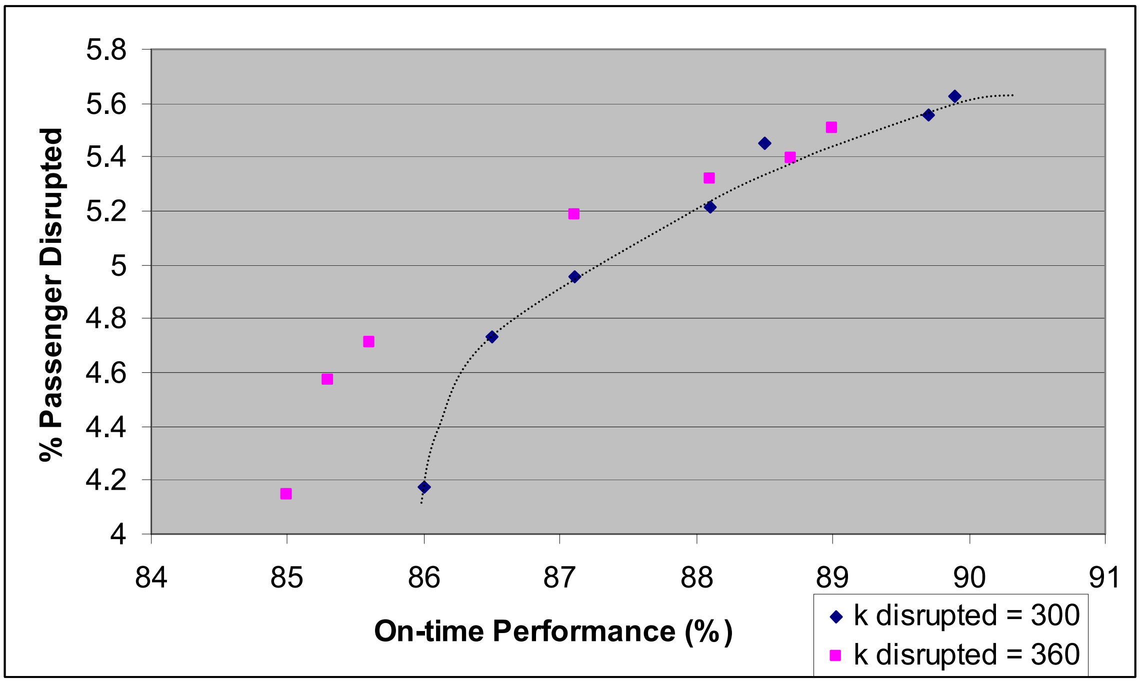

Figure 6.3: % Pax Disrupted vs On-Time Performance, using Proposed Recovery Model

The above plot shows the variation across input parameters and its effect on the 2 objective values. In general as we move from lower on time performance to higher on time performance, there is a corresponding increase of % passenger disrupted. There is obviously a tradeoff involved. In addition, all else being equal, a higher input setting of $k_{disrupted}$ of 360 (more emphasis on passenger disruption) result in a lower % of passenger disrupted. This is true for all the nodes involved.

Since a higher on-time performance and lower % passenger disrupted is desirable, the points forming to the right-most and lower-most corner of the graph would form the pareto optimal solution. The dotted line in the curve above shows the pareto optimal solution plotted out by the combination of input parameter points. Any points to the left and upper corner of the dotted line would, on average, provide a less optimal solution. The following set of (*k_disrupted, d*) provides the pareto front in the graph above:

- (300, 1000)
- (300, 5000)
- (300, 10000)
- (360, 40000)
- (360, 50000)
- (300,40000)
- (300,50000)

An airline schedule recovery crew should ideally pick the solution that is formed out of the above combinations of (*k_disrupted, d*).

6.4 Conclusion

A tentative comparison between the default recovery model and the proposed mathematical recovery model is performed. It is obvious the proposed mathematical recovery model out performs the default recovery model in all the considered performance indexes, at the expense of some additional runtime involved. In addition, a pareto optimality frontier is plotted to achieve % passenger disrupted vs on-time performance. In practice, airline recovery crews often need to make judgment calls based on airline policy to balance these 2 conflicting considerations.

Chapter 7

Simulation Results using Extended Model with Maintenance Consideration

The initial part of this chapter is devoted to detailing the simulation results obtained from SimAir with the extended recovery model proposed. The latter part of this chapter is devoted to discussion and comparison between the two proposed recovery models, and the default recovery model currently existing in SimAir. From the results it is obvious the two proposed models outperform the heuristic rules employed in default recovery algorithm.

7.1 Simulation Settings

A set of SimAir simulation is performed using default recovery model. A second set of SimAir simulation is performed using the extended mathematical model proposed in chapter 5.

The same set of schedule used in proving validity of mathematical model proposed in previous chapter is again used for these 2 sets of simulations.

7.1.1 Airline Legality Rules used for both Simulation

In addition to the passenger legality checks and flight legality checks performed in previous set up, flight maintenance legality check is also performed for the 2 sets of simulation.

7.1.2 Schedule used for Simulation

For the purpose of simulations, the same flight schedule utilized in previous chapter is again used as schedule input. This flight schedule has regular flight maintenance checks of 10 hours apart. Unlike previous simulations, maintenance legality checks are performed for the entirety of the two sets of simulations.

7.2 Results for Simulation Using Extended Model

7.2.1 Simulation Results ran using SimAir Default Recovery

The following simulation results are average results, with SimAir default recovery, collected over 20 iterations of SimAir simulation run.

On average, a total of 10054.73 passengers (approximately 10.47% of total passengers simulated) are disrupted at the end the week. It has an on-time performance index of 86.95%, and an average of 32 flights is cancelled each week.

7.2.2 Simulation Results ran using Extended Model with Maintenance Consideration

SimAir is run using the exactly same setup, schedule and random number seeds to run for the 14 scenarios mentioned in chapter 6. Again, each of the cell values represents aggregate values of simulation over 20 iterations of simulation, each simulation occurring over 1 week's worth of airline schedule. Various disruptions occur during the iteration and recovery module is called into solving the disruption. In this recovery model, there is the added maintenance consideration.

Average On-time performance (%)		
	k_disrupted	
delay cost *d*	300	360
1000	86.4	86.0
2000	86.8	86.1
5000	87.0	86.4
10000	88.0	87.4

20000	88.2	88.0
40000	88.7	88.4
50000	89.3	88.9

Table 7.1: On-Time Performance using Extended Model

Average Passenger Disrupted				
delay cost d	k_disrupted			
	300		360	
	# Pax Disrupted	% Pax Disrupted	# Pax Disrupted	% Pax Disrupted
1000	6334.35	6.60	5278.63	5.50
2000	6380.21	6.65	5310.44	5.53
5000	6430.33	6.70	5374.60	5.60
10000	6910.20	7.20	5566.55	5.80
20000	7582.03	7.90	5662.53	5.90
40000	8061.90	8.40	7006.18	7.30
50000	8271.22	8.62	7254.15	7.56

Table 7.2: % Passenger Disrupted using Extended Model

Average Number of Flights Cancelled		
	k_disrupted	
delay cost d	300	360
1000	32	20
2000	40	22
5000	54	24
10000	93	23
20000	102	53
40000	134	84
50000	145	89

Table 7.3: Number of Flights Cancelled using Extended Model

On average, over 20 iterations, there are 12 disruptions occurring over the one week's worth of schedule. Each disruption would cause simulation to be paused and SimAir would invoke the recovery model 2 to resolve the problem. Each disruption and subsequent recovery would involve 300 flight legs, and take up to 61 seconds to resolve each instance of disruption.

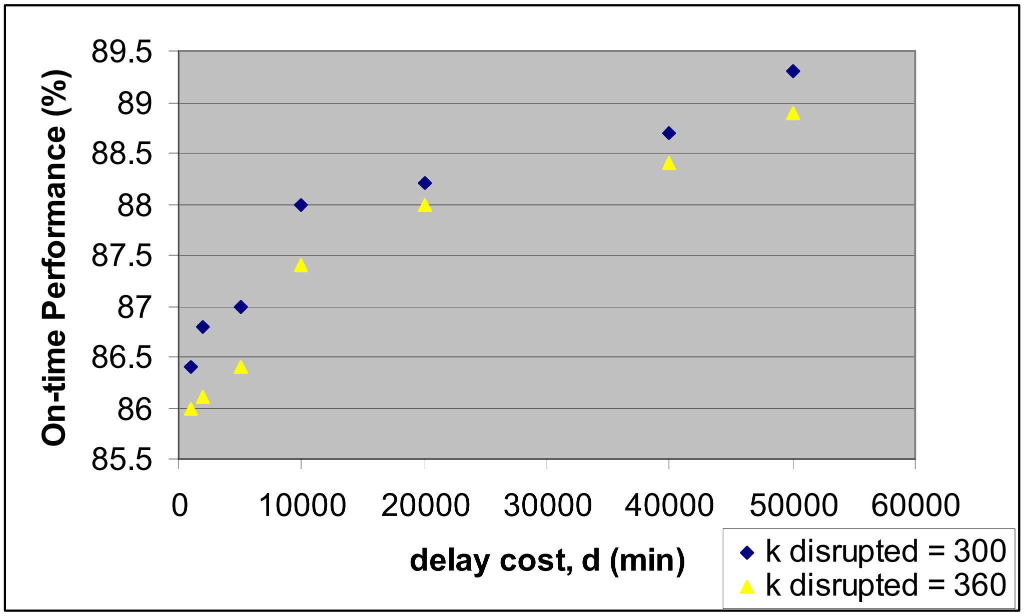

Figure7.1: On-Time Performance vs Delay Cost *d*, using extended model

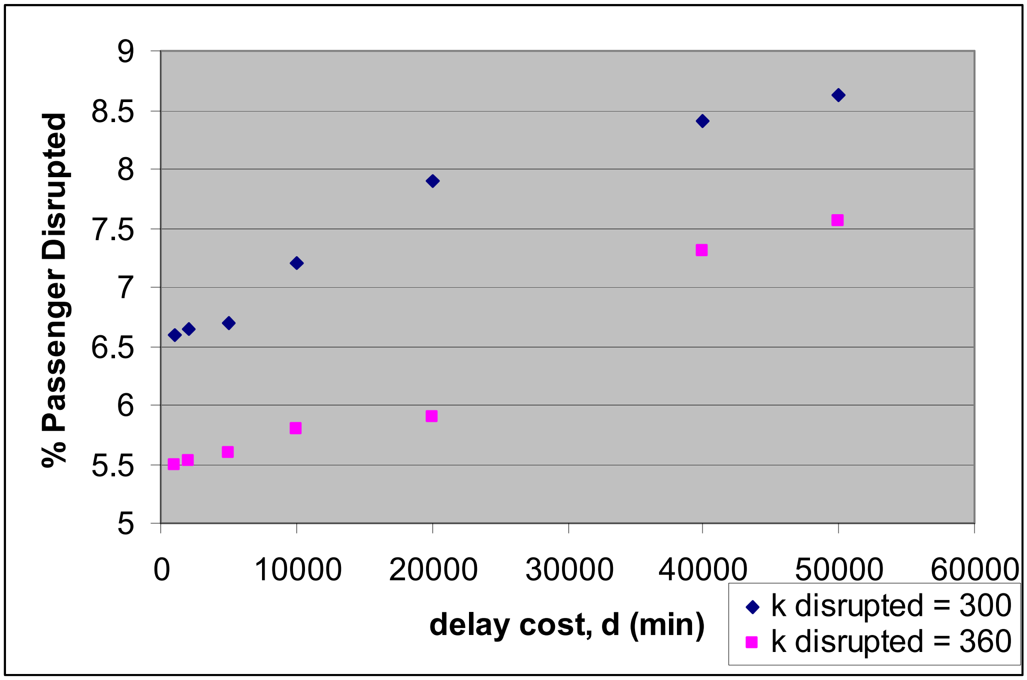

Figure 7.2: % Pax Disrupted vs delay cost, *d*, using extended model

The chart above shows the number of passenger disrupted against delay cost *d*. Two plots are given, with the former with setting $k_{disrupted}$ set at 300, and the latter setting $k_{disrupted}$ at 360.

Similar to our first model, with increasing delay cost d, more passengers are getting disrupted from their original itinerary. Across the two plots, all else being equal, a setting with $k_{disrupted}$ at 360 perform consistently worse than a setting with $k_{disrupted}$ at 300.

There is obviously a tradeoff involved. In addition, all else being equal, a higher input setting of $\mathrm{k}_{\mathrm{disrupted}}$ of 360 (more emphasis on passenger disruption) result in a lower % of passenger disrupted. This is true for all the nodes involved.

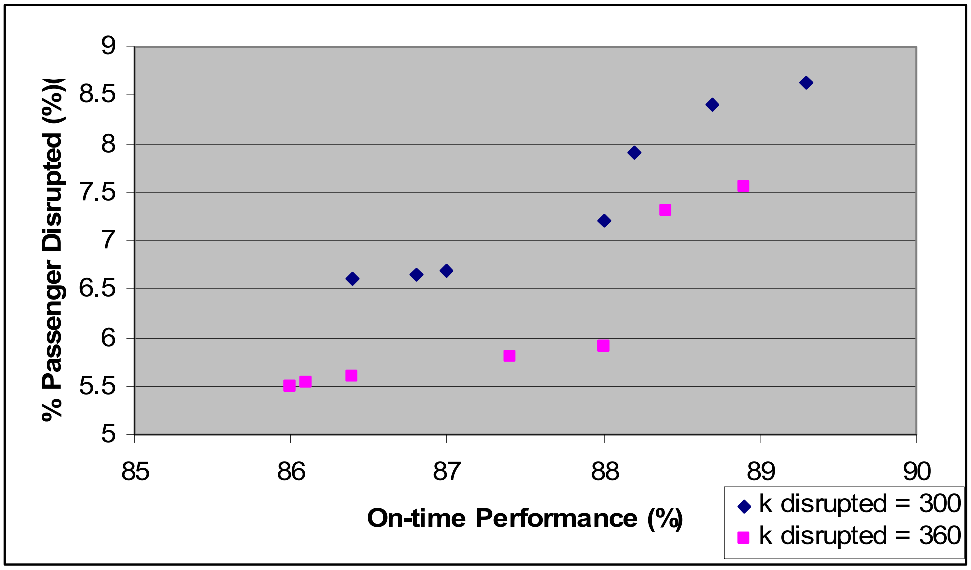

Figure 7.3: % Pax Disrupted vs On-Time Performance, using Proposed Extended Recovery Model

The above plot shows the variation across input parameters and its effect on the 2 objective values. As observed, when we move from lower on time performance to higher on time performance, there is a corresponding increase of % passenger disrupted. Since a higher on-time performance and lower % passenger disrupted is desirable, the points forming to the right-most and lower-most corner of the graph would form the pareto optimal solution. The following set of (*k_disrupted, d*) provides the pareto front in the graph above:

- (360, 1000)
- (360, 2000)
- (360, 5000)
- (360, 10000)

- (360, 20000)
- (360, 40000)
- (360, 50000)

An airline schedule recovery crew should ideally pick the solution that is formed out of the above combinations of (*k_disrupted, d*), should they choose to use the extended model.

7.3 Comparison between the three Recovery Models

The remaining part of this chapter is devoted to discussion on the simulation results of the two proposed recovery models. The results are compared against the default recovery model used in SimAir.

In general it is expected that the second mathematical model performs worse than the first mathematical model. This is understandable since the second model considers maintenance consideration in addition to all the constraints applied to the first model. It is heartening to point out that both models perform admirably better than the default heuristic rule employed by SimAir. The following subsections discuss the various aspects of comparison.

7.3.1 Processing Time

In general, the extended model (with maintenance consideration) takes a slightly longer processing time than the first proposed mathematical model. In comparison, the processing time taken to complete a week's worth of simulation run using default recovery model is almost negligible. All three simulations are ran using a Pentium III 3.0GHz processor.

Processing Time To Resolve An Instance of Disruption (s)	
Default Recovery (chapter 6)	~0
Default Recovery with Maintenance Consideration (chapter 7)	~0
Mathematical Model 1	25.02
Extended Model with Maintenance Consideration	60.9

Table 7.4: Processing Time Comparison

The default recovery is able to find a solution so quickly because internally it utilizes a set of simple heuristic rules. In comparison, the two proposed mathematical models are mixed integer programs and processing time to resolve an instance of disruption is significant. Fortunately, one may consider a processing time of ~1 minute is tolerable in a real life scenario.

7.3.2 Maintenance Violations

The default recovery model in chapter 7 is able to propose a maintenance feasible route for an aircraft during recovery. In comparison, the first mathematical model does not take maintenance consideration into account. It is for this reason that the second extended model with maintenance consideration is proposed.

Average Number of Maintenance Violations over 1 Week's worth of Schedule	
Default Recovery (chapter 6)	2.4
Default Recovery with Maintenance Recovery (chapter 7)	0
Mathematical Model 1	2.4
Extended Model with Maintenance Consideration	0

Table 7.5: Number of Maintenance Violations Comparison

7.3.3 On-time Performance and % of Passenger Disrupted

Total number of passengers disrupted is tallied at the end of every simulation run and an average percentage of passengers disrupted over a week's worth of schedule is calculated.

Average % of Passenger Disrupted (%)	
Default Recovery (chapter 6)	>10%
Default Recovery with Maintenance Recovery (chapter 7)	>10%

Mathematical Model 1	< 6% for all settings
Extended Model with Maintenance Consideration	< 9% for all settings

Table 7.6: Average % of Passenger Disrupted

It is obvious the default recovery fail badly in this regard. This is because the heuristic recovery is concerned with obtaining a feasible and legal flight schedule in face of disruption, and does not take passenger connections into consideration at all. In comparison, the two mathematical models proposed fares much better and have considerably lower passenger disruption over the course of 1 week's worth of simulation.

In general the performance of extended model with maintenance consideration perform slightly worse off compared to original mathematical model with no maintenance consideration. This is illustrated in the two charts below.

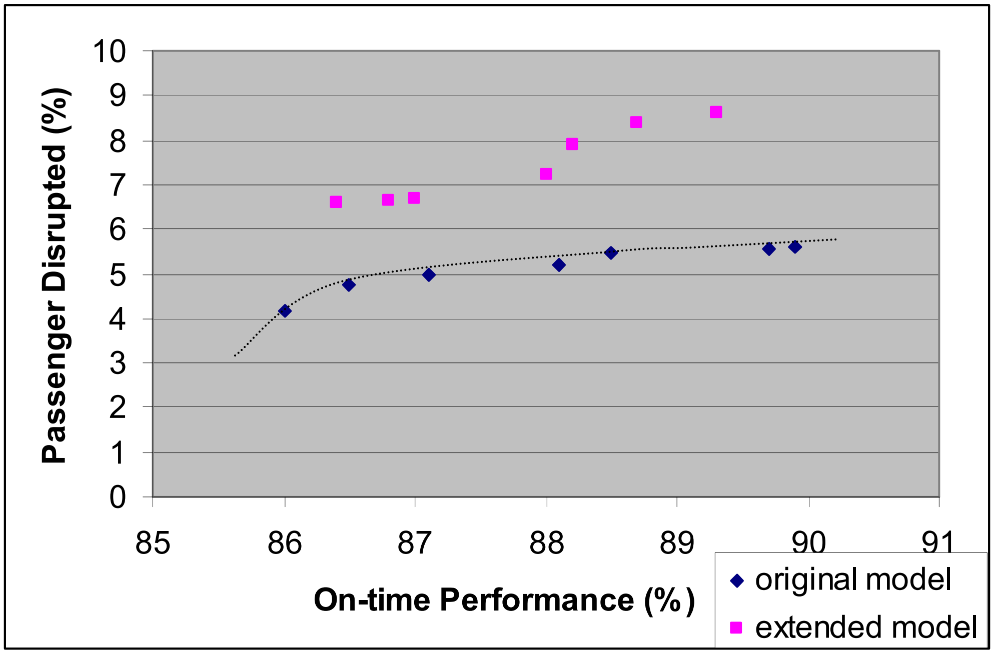

Figure 7.4: % Pax Disrupted vs On-Time Performance, with $k_{disrupted}$=300, using original model and extended model

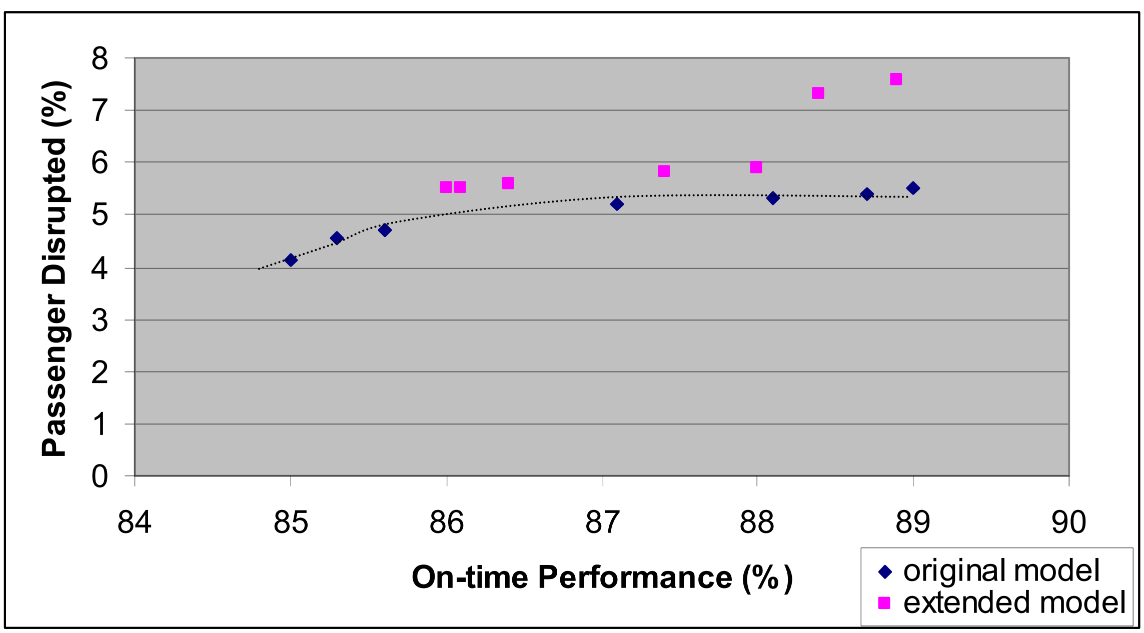

Figure 7.5: % Pax Disrupted vs On-Time Performance, with $k_{disrupted}$=360, using original model and extended model

The first chart summarizes results consolidated with $k_{disrupted}$=300 while the second chart summarizes result consolidated with $k_{disrupted}$=360.

The above 2 charts indicate that in general, all else being equal, the original mathematical model perform better than the extended model. This can be observed using the dotted pareto front drawn out on the two graphs above: the coordinates to the right and lower most positions form the pareto optimal front, and it is consistently dominated by the orginal model.

This is expected since the extended model has the additional maintenance constraint to consider, and is thus has less liberty to choose its possible connections during recovery. With no maintenance consideration included, mathematical model 1 is less constrained and it is hence able to produce a more optimal solution in general.

In all cases, both the models compare favorably with the default recovery employed by SimAir.

7.4 Conclusion

A series of qualitative comparison of the three recovery algorithms have been performed. It is clear that the two mathematical recovery models proposed in this work have demonstrated potentials, and out-performed the heuristic recovery methods employed in SimAir. The computations required of the two models are not excessive either: for the recovery of a moderately sized flight schedule, the runtime on a decent processor only took thirty seconds for the first model, and a minute for the extended model.

In addition, with SimAir as a tool, it helps airlines to perform evaluation on all possible scenarios, and plot out a pareto front to find the best mix of *k_disrupted* and *d* in determining their airline's policy of handling recovery scenarios.

Chapter 8

Conclusion and Possible Future Expansion

This chapter summarizes the findings of this work and suggests a possible expansion to the recovery models proposed.

8.1 Summary and Conclusion

In conclusion, 2 recovery models are proposed in this research. Both models attempt to solve a multi-objective problem of minimizing number of passenger disrupted, and maximizing on-time performance. In particular, recovery model 2 attempt to improve on recovery model 1 by incorporating scheduled maintenance consideration.

Both models, when given a moderately sized schedule, are able to provide optimal solutions within reasonable processing time.

Using an airline simulation model, both recovery models demonstrated their worth and their merit over simple heuristic recovery rules. Both models are able to consider a variety of real world concerns and arrive at an optimal solution quickly.

In addition, together with SimAir, airlines would be able to perform a huge number of simulations to plot out the pareto front to the opposing objectives of meeting ontime percentages and passenger disruptions. From the pareto front plotted, and also the internal airline's policy, one may then pick a point on the pareto front and react to the recovery scenario.

8.2 Thesis Contributions

This thesis contributes to the research of airline schedule recovery in a number of ways:

- An algorithm that generates flight delay options dynamically
    - Presently there are no known work that details the generation of flight delay options in a dynamic manner described in this work
    - The dynamic generation of delay options guarantees that only flight connections that makes sense are generated. This helps to cut down the number of flight connection variables ultimately, and this in turn help to speed up the recovery run time.
- A mathematical recovery algorithm that considers passenger connectivity

    - There are very few published recovery algorithms that looks are passenger recovery and connectivity issue, despite its relevance to real world concern
- An Extended Mathematical Recovery Algorithm that looks at flight maintenance recovery
    - There are also few published recovery algorithms that investigates flight maintenance consideration. This research work attempts to close the gap.
- An Airline Simulation Model that allow researchers to validate usefulness of proposed airline schedule recovery algorithms
    - Again, there are no known airline simulation software that allow researchers to integrate their recovery algorithm seamlessly and easily. SimAir is coded with such a motive in mind.

8.3 Possible Future Research Direction

The proposed recovery model can be extended further to incorporate finer details of maintenance requirements.

While the proposed model (with maintenance consideration), with 2 connection variable type, suffices for most instances of maintenance requirements, there are occasions where recovery fails to satisfy more stringent maintenance requirements. For instance, all that recovery model can guarantee is that the aircraft requiring maintenance before the end of recovery period would get a maintenance slot somewhere before the end of recovery. The eventual scheduled maintenance may be very close to the start of recovery, or very near the end of recovery.

However, there are occasions, albeit rare, where an aircraft requires scheduled maintenance check much sooner. Extended recovery model may schedule a maintenance check much later, resulting in a solution that is infeasible in practice.

There is a way to overcome this problem. One may introduce yet another layer of connection variable type to cater to finer maintenance requirements. This third type of connection variables will have maintenance connection variables that are located much closer to the start of recovery period, and aircrafts requiring sooner maintenance checks must trace out a flight route that passes through the connection variable type 3 described above. The principle is basically the same as recovery model type 2 proposed.

Considering the variety of real world considerations, one can certainly expand upon the recovery models proposed to capture even more constraints. For example, flight crew recovery is not proposed in this work. It is not inconceivable to extend the present work to incorporate the crew recovery considerations.

One may also consider working on passenger recovery by not just minimizing number of disrupted passengers in this work, but also rescheduling and reconnecting the disrupted passengers to alternative flights. However, considering the resultant problem would be rather huge, some compromise would have to be made: it may either be solved sequentially, or some assumptions have to be baked into the recovery algorithm to keep the problem size manageable.

Appendix: References

Andersson, T., V¨arbrand, P. 2004. “The flight perturbation problem”. Transportation Planning and Technology, 27(2): 91 - 118.

Argüello M.F., Bard, J.F. 1997. “A GRASP for aircraft routing in response to groundings and delays”, Journal of Combinatorial Optimization, 1 (3): 211-228.

Benjamin G. T., Bard, J. F., Yu G. 2000. “Balancing user preferences for aircraft schedule recovery during irregular operations”. IIE Transactions, 32:181 -193.

Benjamin G. T., Bard, J. F., Yu G. 2003. “A Bundle Algorithm Approach for the Aircraft Schedule Recovery Problem During Hub Closures”. Transportation Science, INFORMS 37(4): 392-407

Bratu, S., Barnhart, C. 2004, “Flight operations Recovery: New approaches considering passenger recovery”. Working Paper.

Cao, J. M., Kanafani, A. 1997. “Real-time Decision Support for Integration of Airline Flight Cancellations and Delays, Part I: Mathematical Formulation”, Transportation Planning and Technology 20: 183-199.

Cao, J. M., Kanafani, A. 1997. “Real-time Decision Support for Integration of Airline Flight Cancellations and Delays, Part II: Algorithm and Computational Experiments”, Transportation Planning and Technology 20: 201-217.

Clausen, J., Larsen, A., Larsen, J. 2005 “Disruption Management in the Airline Industry – Concepts, Models and Methods”. Technology Report IMM-Technical Report-2005-01, Informatics and Mathematical Modelling, DTU.

Ito, H., Lee, D. 2005 “Assessing the impact of the September 11 terrorist attacks on U.S. airline demand”. Journal of Economics and Business, Elsevier, vol. 57(1), pages 75-95

Jarrah A. I. Z., Yu G., Krishnamurthy N., Rakshit A. 1993. “A decision support Framework for airline flight cancellations and delays”, Transportation Science 27(3): 266-280

Lettovsky, L. 1997 . “Airline Operations Recovery: An optimization Approach”, PhD thesis, Georgia Institute of Technology.

Lee, L. H., Huang, H. C., Johnson, E., Banks, J. 2003. “Discrete Event Simulation Model for Airline Operations: SIMAIR”. Proceedings of 2003 Winter Simulation Conference.

Løve, M. S., K. R., Larsen, J., Clausen, J. 2002. “Disruption management for an airline – rescheduling of aircraft”. Applications of Evolutionary Computing, v2279 Lecture Notes in Computer Science: 315-324. Springer, 2002.

Kohl, N,. Larsen, A.. 2004. “Airline Disruption Management - Perspectives, Experiences and Outlook”. Technical Paper.
http://www.carmen.se/research_development/articles/crtr0407.pdf

Stojkovic, M. Soumis, F.. 2001. “An optimization model for the simultaneous operational flight and pilot scheduling problem”. Management Science, 47(9): 1290-1305.

Talluri, K. 1997 “Swapping Applications in a Daily Airline Fleet Assignment”. Transportation Science 30(3): 237-248.

Teodorovic D., Guberinic S. 1984. “Optimal dispatching strategy on an airline network after a schedule perturbation”. European Journal Operational Research, 15: 178-82.

Teodorovic, D., Stojkovic, G. 1990. “Model for operational airline daily scheduling”. Transportation Planning and Technology, 14(4): 273-285

Yan, S., Young, H. 1996. “A Decision Support Framework for Multi-fleet routing and Multi-stop flight scheduling”, Transportation Research –A 30(5) 379-398

Yan, S. and Lin, C. 1997. “Airline scheduling for the temporary closure of airports”. Transportation Science, 31(1):72 - 81.

Yan, S. and Tu, Y. 1997. “Multifleet routing and multistop flight scheduling for schedule perturbation”. European Journal of Operational Research, 103:155-169.