# MULTI-OBJECTIVE GENETIC ALGORITHM FOR ROBUST FLIGHT SCHEDULING

**TAN YEN PING**

*(B.Eng (Hons.) NUS)*

A THESIS SUBMITTED
FOR THE DEGREE OF MASTER OF ENGINEERING
DEPARTMENT OF INDUSTRIAL AND SYSTEMS ENGINEERING
NATIONAL UNVERSITY OF SINGAPORE
2003

# ACKNOWLEDGEMENT

This research would not have been possible without my supportive supervisors, Dr Lee Chulung and Dr Lee Loo Hay. I would like to thank them for their advice, patience and guidance throughout the two years of my candidature.

Appreciation also goes out to all the professors, research engineers and students in the SimAir team both in National University of Singapore and Georgia Institute of Technology.

I would also like to thank my labmates of Metrology Laboratory, and all the members of the Optimization Research Group (ORG) for making my stay in the ISE department an enjoyable one.

# TABLE OF CONTENTS


**ACKNOWLEDGEMENT** ........ **I**

**TABLE OF CONTENTS** ........ **II**

**LIST OF FIGURES** ........ **IV**

**LIST OF TABLES** ........ **VI**

**ABSTRACT** ........ **VII**

**1 INTRODUCTION** ........ **1**

1.1 Flight Schedule Construction ........ 3

1.1.1 Flight Scheduling ........ 4
1.1.2 Fleet Assignment ........ 4
1.1.3 Aircraft Rotation ........ 5
1.1.4 Crew Scheduling and Assignment ........ 5

1.2 Irregular Airline Operations ........ 6

1.2.1 Recovery Techniques ........ 7

1.3 Trade-off between Robustness and Optimality ........ 8

1.4 Organization of Thesis ........ 12

**2 LITERATURE SURVEY** ........ **14**

2.1 Flight Scheduling ........ 14

2.2 Recovering From Disruptions ........ 16

2.3 Robust Flight Scheduling ........ 18

2.3.1 Insensitive Flight Schedules ........ 19
2.3.2 Flexible Flight Schedules ........ 22

2.4 Evaluating robustness ........ 23

**3 PROBLEM AND MODEL** ........ **24**

3.1 Problem Description ........ 25

3.2 Model Development ........ 25

**4 SOLUTION APPROACH** ........ **32**

4.1 Multi-objective Optimization ........ 32

4.2 Multi-objective Genetic Algorithms ... 35
4.2.1 Genetic Algorithms ... 35
4.2.2 Multi-Objective Genetic Algorithms ... 36
4.3 Components of the genetic algorithm ... 40
4.3.1 Coding Scheme ... 40
4.3.2 Initialization ... 41
4.3.3 Fitness Function and assignment ... 41
4.3.4 Parent Selection ... 43
4.3.5 Crossover and Mutation ... 43
4.3.6 Formation of Child Population ... 46
4.3.7 Handling constraints and infeasible solutions ... 46
4.4 Overall procedure ... 47

**5 SIMULATION STUDY ... 52**

5.1 Overview of SIMAIR 2.0 ... 53
5.1.1 Simulation module ... 54
5.1.2 Controller Module ... 57
5.1.3 Recovery Module ... 59
5.1.4 Performance Measures ... 61
5.2 Measure of Robustness ... 62
5.2.1 Operational FTC ... 62
5.2.2 Operational Percentage of Flights Delayed ... 64
5.3 Test Data ... 64
5.3.1 Generating the Flight Schedule and Aircraft Rotation ... 65
5.3.2 Generating the Crew Schedule ... 65
5.4 Parameter Setting ... 69

**6 RESULTS ... 71**

6.1 Test Data *A* ... 71
6.1.1 Non-dominated front ... 74
6.1.2 Performance of percentage of flights delayed ... 76
6.1.3 Performance of operational FTC ... 82
6.2 Test Data *B* ... 83
6.2.1 Test Data *C* ... 85
6.3 Summary ... 87

**CONCLUSION ... 88**

**REFERENCES ... 1**

# LIST OF FIGURES

Figure 1.1 Decomposition of the elapsed time of a duty ............................................... 6

Figure 4.1 A population of five solutions ..................................................................... 33

Figure 4.2 Approaching the Pareto front for a two-objective problem......................... 34

Figure 4.3 Pareto ranking of a population of solutions................................................ 38

Figure 4.4 Fonseca's method of ranking solutions of a multi-objective problem ........ 39

Figure 4.5 Update of the child and elite population..................................................... 51

Figure 5.1 An overview of the operational SIMAIR model ........................................ 53

Figure 5.2 Decomposition of a leg................................................................................ 55

Figure 5.3 Graphical representation of flight network used in test data ....................... 64

Figure 5.4 Time representation of flight schedule used in test data *A* .......................... 66

Figure 5.5 Time representation of flight schedule used in test data *B* .......................... 67

Figure 6.1 Movement of elite population towards the Pareto front over several generations of the Genetic Algorithm ................................................................. 72

Figure 6.2 Elite Population of generations 300, 500 and 700 of the Genetic Algorithm (for test data A) ............................................................................................. 73

Figure 6.3 Comparing solutions in elite population 300 with the original flight schedule.............................................................................................................. 74

Figure 6.4 Rotation 002 of test flight schedule ........................................................ 75

Figure 6.5 Comparing the average delay of flights in rotation 002 ............................ 77

Figure 6.6 Improvement in the delay of flights ........................................................ 78

Figure 6.7 Percentage of flights delayed against the delay in minutes (Top). Cumulative percentage of delay in minutes for different solutions (Bottom) ...... 80

Figure 6.8 Comparing the shift between the original schedule and the improved schedule for crew pairing 1907 ........................................................................ 81

Figure 6.9 Progression of elite population for Test Data *B* ........................................ 84

Figure 6.10 300th Elite population for original test data and test data B .................... 85

Figure 6.11 Progression of the elite population for Test Data *C* ................................. 86

# LIST OF TABLES

Table 5-1 Parameters used in the 8-in24 hours rule ..................................................... 58

Table 5-2 Crew structure for test data sets................................................................... 67

Table 5-3 Values of Parameters used in solution procedure......................................... 69

Table 6-2 Sequence of flight in rotation 002 ................................................................. 76

Table 6-2 The sequence of flights in crew pairing 1907 .............................................. 82

Table 6-3 Set of Parameters used to compute FTC ...................................................... 68

Table 6-4 Computation of each duty cost in pairing 1907 for Test Data *A* .................. 69

# ABSTRACT

Traditional methods of developing flight schedules generally do not take into consideration disruptions that may arise during actual operations. Potential irregularities in airline operations, such as equipment failure and baggage delay are not adequately considered during the planning stage of a flight schedule. As such, flight schedules cannot be fulfilled as planned and their performance is compromised, which may eventually lead to huge losses in revenue for airlines.

In this thesis, a procedure to improve the robustness of an existing flight schedule was developed. The problem is modelled as a multi-objective optimization problem, optimizing the departure times of flights, allowing airlines to improve on more than one objective. The procedure developed to solve the problem is built on the basics of multi-objective genetic algorithms. A simulation model, SimAir, that models the operational irregularities has been employed to evaluate the performance of the flight schedule. SimAir considers different performance measures (or criteria) such as flight cancellation, operational cost and other performance indices as well.

# 1 INTRODUCTION

Air transport is the fastest growing transport industry with air passenger traffic growing an average yearly rate of 9% since 1960. It has become a major service industry contributing to both domestic and international transport systems. Air transport facilitates widen business communications and is a key component in the growth of tourism, now one of the world's major employment sectors.

One of the strong sources of income for airlines is the business travellers who are willing to pay up to five times for a ticket as compared to the rest. This accounted for 10% of the industry's passenger volume and 40% of its revenue. But this group of people began to opt for low-fare carrier in the late 1990s; cheaper flights from discounters came into favour. As the business traveller base began to shrink and the economy began to slow down in early 2001, operating cost became a greater burden for major airlines. In the near future, the route networks of low-cost airlines might grow large enough to make alternative service available in almost all of the large business markets. To make things worse, the September 11 attacks deterred travellers from flying. With regards to United Airline's recent file for bankruptcy, Aaron Gellman an aviation expert at Northwestern University believes that United Airlines will emerge from bankruptcy and they'll come up leaner and meaner as a competitor. This shakeup may ripple across the industry, leading to competitive cost-cutting among airlines. Competition from low-cost airlines, terrorism and other factors are forcing U.S. major hub-and-spoke carriers to restructure their operations improving their efficiency or face the prospect of eventually going out of business.

The prospects of the aviation industry in Asia have also been bleak. The air travel in year 2001 fell sharply as a result of the slowdown in the world's major economies;

exacerbated by the September 11 attacks in the US. Along with Cathay Pacific and Qantas, Singapore Airlines has been one of the most profitable carriers in the world. But it was hit hard by the October 2002 terrorist bombing in Bali, and suffered further setbacks from the conflict in Iraq. The outbreak of SARS in March this year brought added pressure on airlines that report sharp falls in bookings and are being forced to cut back flights. Singapore Airlines said it was cutting 125 flights a week in response to its falling demand. Even after reduction of its services, Singapore Airlines announced that a further move to retrench cabin trainee and other operations staff.

Nothing is more basic to an airline than the flight schedule it operates. Since every instance of a flight schedule affects the revenue of an airline, they are of paramount importance for every airline. As such, constructing a quality flight schedule is essential to the airline. Developing airline flight schedule is a very intricate task. Current state of the art optimization techniques generate highly resource utilized and hence efficient schedules. Consequently, airlines operate on highly optimized tight flight schedules. These flight schedules are tightly woven, highly interrelated structure of legs. Many aspects are rigidly governed by specific regulatory or contractual requirements, such as those relating to maintenance of equipment, and working conditions of flight crew. Moreover, almost every schedule is inter-wined with other scheduled flights because of connections, equipment routing and other factors. A major, yet unrealistic assumption made when modelling the problem of constructing the flight schedule is to assume that the airline operations are deterministic, i.e. they plan flight schedules assuming that they will be performed as planned, without consideration of the potential delays and unexpected external events. However, from Rosenberger (2001a), it is seen that schedules are in reality frequently disrupted by unplanned external events such as bad weather, crew absence or equipment failure. When an unforeseen event occurs, causing a delay in the first flight of the day, without sufficient slack time between flights, this delay may propagate along the flight network to the rest of the flights that are flown be the aircraft and crew, causing wide spread disruption in the system. Passengers missing their connection due to delay may lose goodwill towards the airline. It was reported in The Atlanta Journal-Constitution (2002) that weather is responsible for about two-thirds of all delays. These disruptions occur every single day in airline operations, consequently, in 2001, only 73.4% of the flight arrived on time and up to 3.87% of flights were cancelled (BTS, 2002).

One challenge of the flight scheduling process is to be able to build a schedule that is robust such that it will be able to perform relatively well under various operational irregularities, be it harsh weather conditions or equipment fault.

## 1.1 Flight Schedule Construction

The flight schedule represents one of the primary products of airlines. An airline has the responsibility to provide adequate service to the cities it serves; it must also, operate efficiently and economically. Therefore, in its scheduling practices, airline management must continually search for the balance between adequate service and economic strength for the company.

Airline flight scheduling includes all the planning decisions that have to be made for a schedule to be considered operational. It normally consists of the scheduling of aircraft maintenance, route development for the aircraft and crew scheduling. Flight Airline operations are made up of many interdependent components, making the planning problem a very complex problem to be solved. Besides meeting the customers demand, the airline has to incorporate into their planning many other constraints pertaining to the airport facilities, seasonal considerations, aircraft maintenance and crew members.

The produced schedule not only has to comply with all the Federal Aviation Administration (FAA) rules that require all the aircraft to receive periodic maintenance, it also has to satisfy the union agreement allowing crew member to have a minimum amount of rest.

To handle the complexity of the problem, the usual approach to planning the airline schedule is to decompose the overall problem into sub-problems and solving these sub-problems independently with various optimization techniques. These sub-problems have been well studied and many linear optimization techniques have been developed to solve them individually. By solving the sub-problems sequentially, a preceding sub-problem delivers the input data for the subsequent sub-problem. Wells (1999) discusses each of the components of airline scheduling in detail; only issues relevant to this study are discussed here.

### 1.1.1 Flight Scheduling

Flight schedules are commonly constructed based on market demand. Historical data about bookings from computerize reservation systems are utilized to perform traffic forecasts for each origin-destination pair. The result of market evaluation is used to generate the flight network and assign frequency to the legs. Flight scheduling determines the origin, destination, departure time and arrival time of each flight.

### 1.1.2 Fleet Assignment

Once the flight schedule is in place, fleet assignment is carried out. A *fleet* is a collection of aircraft that is of the same aircraft type. A separate maintenance-routing plan must be drawn up for each type of aircraft in the fleet; this is essentially what is accomplished in fleet assignment. Maintenance of airplanes requires that certain stations be provided with facilities and personnel for periodic mechanical checks. All routing plans must be coordinated to provide the best overall service.

### 1.1.3 Aircraft Rotation

Airline planners refer to a specific aircraft by a *tail number*. An aircraft rotation is an ordered sequence of legs that can be assigned to tail number. At the end of the aircraft rotation problem, tail numbers are assigned to the rotations. For safety reasons, aircrafts must be regularly maintained, thus, maintenance must be embedded within the aircraft routes. Also, there should be adequate turn time for the aircraft, that is to say that when an aircraft arrives at that gate, there should be sufficient time for the ground personnel to service the aircraft and transfer baggage before the plane leaves for its next leg; also, the passengers need time to move out of the plane and they have to allow time for the next group of passengers to move in. With the available set of aircrafts, airlines deal with the rotation problem through maximizing aircraft utilization.

### 1.1.4 Crew Scheduling and Assignment

On completing the aircraft rotation, airlines solves the crew scheduling problem. The crew scheduling problem partitions the set of legs into pairings (or trips) that crews will fly. The crew is fleet type specific; pilots are usually qualified for one aircraft type only. A *crew pairing* is a sequence of flights originating and terminating at the same crew base. A crew pairing is made up of a sequence of duties; a *duty* is a set of legs flown by a crew in a day. The duration between the start of a duty and the end of a duty is the *elapsed time*, it includes a briefing period before the first leg of the duty and a debriefing after the last leg of the duty. An example of the decomposition of the elapsed time of a duty is illustrated in Figure 1.1.

Crews may only fly for a certain number of hours in a day, week and month. They must also have sufficient time to transfer from aircraft to aircraft, and have adequate overnight rest. Every pairing is constructed so that a single crew can legally perform all the work activities it contains. After fuel costs, crew costs are the highest direct operating cost of an airline. It was report that American Airlines pays about US$1.3 billion in salary and benefits to 8300 pilots. Thus, crew pairings are scheduled to maximize crew utilization while conforming to the numerous contractual restrictions from the union.

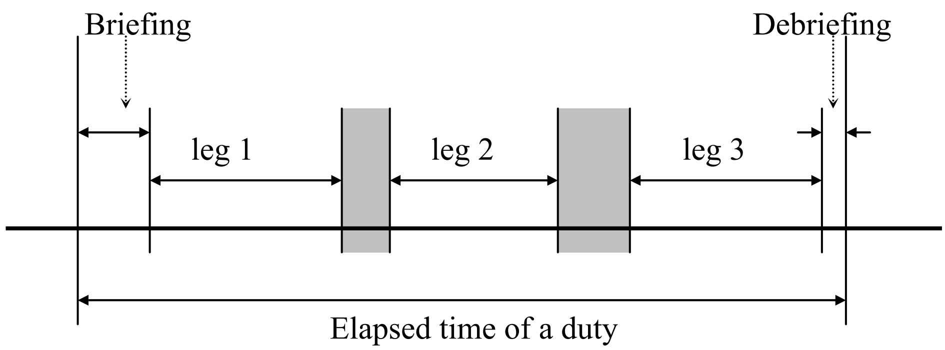



**Figure 1.1 Decomposition of the elapsed time of a duty**

The constructed crew pairings are then assigned to each individual crew. This is usually done using a bidline model. A *bidline* is a set of pairings that a crew flies within a month. A set of bidlines are generated and the pilots sequentially choose the bidline they prefer in order of seniority.

## 1.2 Irregular Airline Operations

Airline operations are subjected to a high level of uncertainty arising from numerous factors. These factors that cause disruptions to the operations ranges from inclement weather conditions, equipment failure, and crew unavailability to baggage delay. Any condition that prohibits the airline from operating the flight schedule as planned is considered as a disruption.

### 1.2.1 Recovery Techniques

These disruptions brought about by various factors can upset the entire flight schedule. Snow, thunderstorm and other forms of bad weather can lead to degradation in the airport's capability to handle aircrafts that are taking off and landing from it; in worst cases, the airport is forced to close down for a short duration. To reduce the impact brought upon by irregularities, a common approach is to develop real-time techniques that can be used to re-optimize the schedule when a disruption occurs. These techniques are commonly known as *recovery techniques*. The basic and most common objective of recovery is to reduce the impact of the disruption on the rest of the flight schedule. It is usually accomplished by assigning costs to flight cancellation and passenger delay, and minimizing the combined cost in hope that the new schedule suggested by the recovery procedure would adhere to the original flight schedule as much as possible. More often than not, airlines are forced to make drastic decisions such as cancelling flight legs or delaying flights for long durations in an effort to recover back to the original schedule. However, these decisions prove costly to the airlines.

When flights are cancelled in a recovery attempt, aircraft rotation will be changed. The new set of aircraft routing have to satisfy all maintenance requirements. If cancellation is not possible, the recovery searches for a chance to swap parts of aircraft routing of the disrupted aircraft with that of other aircrafts. If a spare aircraft is available at the airport of the problem aircraft, a substitution can be made. The last resort would be to ferry aircraft between stations. Ferrying an aircraft is simply flying it without passengers. It can be performed on an aircraft that is 'stuck' at an airport without the required maintenance facilities by flying it to a suitable station. Ferry is also done on a spare aircraft that is required to replace one that is out of service at another station. Ferry is used last the last option as no revenue is generated and a crew must be paid to fly the aircraft.

When a decision to cancel flights is made, the passengers who were supposed to fly on the cancelled flights have to be reviewed. New itineraries have to be created for these disrupted passengers. In an event of misconnections, passengers might get stranded at an airport for the night. For these passengers, the airlines have to compensate them for their accommodation.

In the midst of a disruption, a crew might be unable to connect to his next flight. In such a situation, airlines would commonly call upon a reserve crew to replace this disrupted crew. However, this kind of recovery is very expensive to the airline. Not only does the original crew gets paid for the next flight that he missed, the airline has to pay for the reserve crew that was utilized.

Equipment failure and bad weather conditions are not within the control of the airlines, thus recovery policies and models that are able to solve the problem in a short time have to be designed to reduce the impact of these disruptions. Without proper recover policies in place, subsequent legs along the network might also be affected. Statistics of every flight whether it is cancelled or on-time is published regularly. On-time performance leads to a higher customer satisfaction and plays a major role in the airline become the carrier of choice.

## 1.3 Trade-off between Robustness and Optimality

Judging from the high rate of delays and cancellations, it is clear that in addition to generating an optimized flight schedule, one has to be concerned with the robustness of this schedule operating in the real world, with its accompaniment of unexpected yet frequent disruptions.

It is necessary to recognize that there is a trade-off between robustness and optimality of a flight schedule. A robust flight schedule usually will not correspond to the optimum of the objective function of the airline schedule planning problem. However, given that a robust schedule can better withstand disturbance, it does not mean that such a schedule will bring in lesser profits for the airline, or will be inferior when subjected to operations. On the other hand, a very efficient flight scheduling solution might be optimal in a deterministic environment, but highly unreliable (and thus sub-optimal in some criteria) when implemented in a daily operational environment.

Robustness of a flight schedule can be broadly classified into two categories. The first category is the degree of the flight schedule's insensitivity to external disturbance. In other words, a flight schedule is considered robust if it will not badly affected when different forms of disruptions occur. A list of measures that can be used to measure the insensitivity of a flight schedule is given below.

- On-time performance. A leg is considered on-time if it arrives at the gate within 15 minutes of its originally scheduled arrival time. The on-time percentage is the percentage of the number of on-time legs as a percentage of the number of legs schedule. A cancelled leg is considered as not on-time. On-time performance is a measure of the adherence of the flights to its original schedule.

- Percentage of flights delayed. This measure is usually partitioned into two categories, percentage of late departures and percentage of late arrivals. A flight is considered late if it departs / arrives after 15 minutes of it scheduled departure / arrival time. A cancelled leg is also considered late. This percentage serves as a measure of timeliness.

- Average minutes late for each flight in the schedule over a period of time. Like the on-time percentage and percentage of flights delayed, the average minutes late for a flight is an indication of how well the flight schedule performs in operations, and its ability to adhere to the originally planned schedule.

- Number of legs cancelled per day. Legs are cancelled by a recovery procedure as a result of disruption. Cancelling legs is a costly process, with leg cancellation, passengers have to be re-accommodated on other flights or other airlines. Hence, airlines need to keep this number to the minimum.

- Average number of disruptions in a day that result in the need for an aircraft / crew / passenger recovery procedure. Different disruptions require different forms of recovery; for instance, if an aircraft unexpectedly runs into a minor equipment failure, a short delay of flight is sufficient to solve the problem without the need to modify the crew plan or put the passengers on other flights. Another disruption example is when the airline realizes that the crew that is needed to fly a leg is delayed due to a previous flight; the airline can call in a reserve crew without disturbing the rest of the plan. However major disruptions can occur, such as an airport closure can lead to the need for all three forms of recovery. This measure, thus keeps records of disruptions that result in the need for different types of recovery.

- Operational crew cost. Crew cost is one of the highest operational costs of an airline, thus it is essential for the airlines to be able keep the crew cost down. Different airlines employ different pay structures of the crew. A typical structure used by most American airlines is the flight-time credit (FTC). The definition of FTC is provided in a later Chapter.

- Number of crew violating a maximum block-time rule. For example, many airlines use an 8-in-24 rule, which states that a crew should not fly more than 8 hours in any 24 hour window. This measure reflects the tightness in a crew schedule. If this rule is always violated, the airline might have to look into adding some form of slack to the crew schedule so as to bring this value down.

- Number of reserve crew required to cover the duties of a disrupted crew. One form of crew recovery is to call upon a reserve crew to replace a regular crew when the crew is unavailable. However, by doing so, both the regular crew and reserve crew will be paid.

- Percentage of disrupted passengers. A passenger is considered disrupted if he did not fly his itinerary on the original scheduled flight, i.e. he is rerouted or the flight is cancelled. This measure is important to the airlines as passengers that are disrupted might lose interest in the airline and make a switch to other airlines.

- Percentage of inconvenienced passengers. A passenger is considered inconvenienced if his flight is delayed for more that a certain amount of time. In the same way as the percentage of disrupted passengers, this measure is important to the airlines as a measure of providing good service.

## 1.4 Organization of Thesis

This thesis focuses on the problem of incorporating operational considerations into the airline schedule planning process. It takes the approach of reducing the schedule's sensitivity to irregularities that are frequent in operations. Instead of developing a new model for airline scheduling, the problem seeks to improve the robustness of an existing flight schedule. To evaluate the robustness of a flight schedule, simulation is performed.

In chapter 2, a survey of the past literature on common approaches to flight scheduling, recovery and robust airline schedule planning is documented.

Chapter 3 describes the motivation behind this research project and defines the problem that can be solved to improve the robustness of flight schedules in detail. Robustness of flight schedules can be measured by means of various criteria. Often, airlines wish to improve on more than one criterion when planning their flight schedule; hence, the problem is formulated as a multi-objective problem.

Chapter 4 details the Multi-objective Genetic Algorithm (MOGA) procedure that is developed to solved the problem that was described. Principles of multi-objective optimization with traditional ways of dealing with such problems are discussed. It also provides an overview of genetic algorithms and how it is applied to multi-objective problems.

Chapter 5 describes the simulation model (SIMAIR) used to evaluate each of the new flight schedules generated by the procedure and the statistics that are collected by the simulation program.

Chapter 6 outlines the results of this research project by applying the procedure to a flight schedule. It is shown that the solution procedure can improve the robustness of a flight schedule by a significant amount.

Chapter 7 summarizes the ideas that were introduced in this project, and discusses possible directions for future research in this area.

# 2 LITERATURE SURVEY

In the last two decades, substantial research has been conducted on airline schedule planning. Most of them decomposed the enormous problem into sub-problems optimizing them independently while others integrated one or more of the sub-problems. However, very little research has been done on the problem of addressing the impact of irregular operations, and developing models that will result in robust flight schedules that are less sensitive to operational disturbance. A majority of studies that were carried out dealt with irregularities on a different note; they developed models and decision support systems to handle the problem of disruption only when it occurs, instead of building robustness into their original schedule.

In this chapter, a brief review of models used to plan different stages of flight scheduling is outlined. Methods and policies that studied to help an airline recover from disruptions are also described. Finally, previous research conducted other researchers on robust flight scheduling is presented; these studies take into account the effects of disruptions in the planning stage.

## 2.1 Flight Scheduling

Flight schedule planning and in particular, crew scheduling have long been the most success applications of Operations Research.

The fleet assignment model problem is of considerable importance to airlines, much of research have been done to solve the daily fleet assignment problem optimally. Abara (1989), Hane *et al*. (1995) and Subramanian *et al.* (1994) presented models to solve the daily fleet assignment problem; minimizing a combination of operating cost and the opportunity cost of spilling passengers.

Clarke *et al*. (1996b) extended the daily fleet assignment problem to provide modelling devices for including maintenance and crew considerations into the basic model while retaining its solvability. In this model, only maintenance checks for short durations are considered. Assuming that the fleet assignment problem is solved, Clarke *et al*. (1996a) also developed a model that solves the aircraft rotation algorithm to determine the routes flown by each aircraft in a given fleet.

In Sriram and Haghani (2003), the author's fleet assignment model explicitly caters to maintenance scheduling for both short and long maintenances. The objective is to minimize the maintenance cost and any cost incurred during the re-assignment of aircraft to the flight segments. The model is solved using a heuristic approach.

Combining the fleet assignment problem and the aircraft rotation problem, Barnhart *et al*. (1998) presented a model and solution approach that can be used to solved the problem in a single step. Cost associated with aircraft connections and maintenance requirements are captured in the model and it is solved by a branch-and-price solution approach.

Over the years, a considerable amount of work has been produced by operational researchers on crew scheduling. The most common approach centred on modelling it as a set-partitioning problem. To use such a formulation, pairings must either be enumerated or generated dynamically; it can be a complex task due to the numerous legality rules enforced. Hoffman and Padberg (1993) found optimal integer solutions to problems with a maximum of 300,000 pairings using a branch-and-cut algorithm. In their approach, crew base constraints were explicitly considered.

Graves *et al*. (1993) describes the crew scheduling optimization system used by United Airlines. The system uses a variation of set partitioning formulation to find an initial feasible solution by allowing flights to be overcovered or uncovered by paying a penalty. Once an initial feasible solution is found, local optimization is used to find potential improvements.

Vance *et al*. (1997) presented a different model for airline crew scheduling, based on breaking the decision process into two stages. The first stage selects a set of duty periods that cover the flights in the schedule and the second builds pairings using those duty periods.

Conventionally, each stage in the scheduling process was treated as an independent problem. However, we must not overlook the fact that there is a high degree of interdependence between stages; by constructing it stage by stage and optimizing different objectives at each stage, there is no strong basis to show that the flight schedule and plan that has been developed through the stages will be optimal as an entity. Thus in recent years, there have been attempts to solve several stages of the planning process together. Grosche *et al*. (2001) developed an integrated, GA-based flight schedule construction approach which simultaneously permits multiple planning activities like airport selection, leg selection, departure and arrival scheduling, aircraft rotation and fleet assignment. The flight schedule is represented as a list of flights with departure station and time. Langerman *et al*. (1997) proposed an agent-based airline scheduling procedure integrating the different components of airline scheduling. The proposed model used to develop schedule is market driven with maintenance and crew requirements as constraints.

## 2.2 Recovering From Disruptions

As airlines have done a better job solving fleet assignment and crew scheduling to optimality, flight schedules become more optimized, with minimal slack between flights, making it more susceptible to disruptions. This has led to an increased need for recovery methods that can be employed in an event of disruption. Researchers began to develop recovery models and decision support systems to deal with unexpected disruptions.

Teodorvic and Guberinic (1984) were the first to publish an aircraft recovery model for minimizing the total passenger delay. The same authors then extended their work to allow cancellation and include the airport operating hours. The problem is formulated to define a new daily flight schedule (aircraft routing), when one or more aircraft is taken out due to a disruption. They attempted to find the least expensive set of aircraft routings using a branch and bound procedure.

Jarrah et al. (1993) presented two minimum cost network flow models to incorporate delay and cancellation. The objective is to systematically adjust aircraft routing and flight scheduling in real time to minimize total cost incurred from a shortage of aircraft.

Yan and Yang (1997) first combined cancellation of flights, ferrying of spare aircraft and delays of flights in a single model for aircraft recovery. The problem was represented using a time-space network. To minimize the duration of schedule perturbation, a simple decision rule is used. This framework was extended by Yan and Lin (1997) to handle station closures.

Thengvall et al. (1998) approached the aircraft recovery problem in a way that allows an airline to provide for schedule recovery with minimal deviations for the original aircraft routings. A network model with side constraints is presented in which delays and cancellation are used to deal with aircraft shortages in a way that ensures a significant portion of the original aircraft routings remain intact. The same authors also developed multi-commodity network-type models for determining a recovery schedule for all aircraft (multiple fleets) operated by a large carrier following hub closure (2001). The models allow for cancellations, delays, ferrying and substitution between fleets and sub-fleets.

Rosenberger et al (2001c) presented an optimization model that reschedules flight legs, and reroute aircraft by minimizing an objective function involving rerouting and cancellation costs. The author also developed a heuristic for selecting which aircraft to be rerouted.

Although there is an extensive literature on Airline Crew Scheduling, studies on crew recovery during irregular operations are few. Teodorovic and Stojkovic (1995) developed a sequential approach based on a dynamic programming algorithm, using first-in-first-out principle to minimize the crew' ground time. Lettovsky et al. (2000) presented a new solution framework to reassign crews and restore a disrupted crew schedule. Pre-processing techniques are applied to extract a subset of the schedule for rescheduling. A fast crew pairing generator is built that enumerates feasible continuous of partially flown crew trips.

Lettovsky (1997) also formulated an Airline Integrated Recovery (AIR) model for optimal recovery from schedule disruption. The model includes variables and constraints pertaining to all three aspects of a flight plan (crew assignment, aircraft routing and passenger flow) for the problem for a given airline. The solution algorithm is derived by applying Benders' decomposition algorithm to a mix-integer linear programming formulation for the problem.

## 2.3 Robust Flight Scheduling

Current flight scheduling models are planned in a deterministic environment; they define their objectives mainly on costs, resulting in schedules that are unable to withstand disruptions. As such, recovery has to be summoned each time a disruption occurs. Robust flight scheduling is to take into account operational irregularities during the planning stage so that it is less sensitive to disruption or it can better recover in the occurrence of disruption.

Robustness of a flight schedule can be assessed in many different ways, for instance, a flight schedule that results in a minimum overall flight delay might be a measure of how robust a flight schedule is. Due to the numerous different ways of assessing the robustness of a flight schedule, there is no common basis for researchers to build on; this might be a probable reason to why robustness of flight schedule was not investigated upon until the recent couple of years. Most of the studies conducted on constructing robust flight schedules focused developing models for either the crew or the aircraft only, instead of considering the entire planning process.

### 2.3.1 Insensitive Flight Schedules

A group of researches define the robustness of flight schedules as the amount of insensitivity of the flight schedule to external disturbances. By this definition, robustness can be assessed in many different ways, some of which was provided in the previous chapter.

Barnhart (2001) aimed to develop a robust schedule pertaining to fleet assignment. The author assessed the robustness of a fleet assignment model using the schedule's impact on delay and cancellation. The robust fleet assignment model (RFAM) that was developed is adapted from aircraft recovery model developed by the same author. The aircraft recovery model's objective is to determine which flights to cancel and at what time the remaining flights should depart so as to minimize delay and cancellation costs. Thus, in the author's RFAM model, the goal was to build paths covering the time period's work which are optimal with regards to penalties for delayed arrivals, penalties for cancelled flights and penalties for fleet imbalances at the end of the day. Barnhart (2001) also showed a list of metrics that can be used to compare if one flight schedule is better than another (in terms of its insensitivity) , categorized according to different aspects of the airline operations, namely the aircraft, crew and passenger.

Listes and Dekker (2002) made a study on robust fleet composition to determine the number of aircraft of each type the fleet should consist of in order to be most profitable when assigned to a schedule. The author's idea was to search for a fleet composition which appropriately supports dynamic allocation, depending on the flight schedule under construction and the associated stochastic demands on its flight legs. The main measure of fleet performance is expressed in terms of the profit it can generate by operating the schedule from which the fixed costs of its aircrafts have to be subtracted.

Wu and Caves (2002) developed a model to optimize the scheduling of aircraft rotation by balancing the use of schedule time, which is designed to control flight punctuality, and delay costs. The model seek to determine the optimal schedule buffer time at airport and block times between airports minimizing system costs in aircraft rotations by optimizing the allocation of schedule buffer time in aircraft rotation schedules. Adherence of the schedule implementation to the planned schedule i.e. mean delay time of aircraft rotation, expected delay time of aircraft rotation and schedule regularity are employed to evaluate the reliability of aircraft rotation schedules.

Yen and Birge (2000) models the crew scheduling as a stochastic problem by explicitly including the cost of disruptions in the scheduling formulation. The delay cost is added to the deterministic objective function in order to take into account delays that affect flight segments constraints. By doing so, stochastic disruptions (short range effects) are considered in the long range crew scheduling problem. The model also captures the interaction and interdependence between crew assignments by using a two-stage stochastic program with recourse. The authors have shown that significant savings can be achieved if delay effects on crew schedules, which consequently affect the entire system, are considered during the planning phase.

Schaefer et al. (2001) seek better approximate solution methods for crew scheduling under uncertainty that still remains tractable. The author noted that airlines have traditionally evaluated a crew schedule by its planned cost; his method of evaluation is to determine the operational cost which is obtained through simulation. Two methods were developed to find robust crew schedules. The first method minimizes the expected crew cost by considering each pairing in isolation. The next method is a penalty method that penalizes certain attributes in a pairing that may lead to poor performance in operations; for example, if the maximum duty duration is near its limits, a penalty cost is added to the scheduled pairing cost that is to be minimized.

Ehrgott and Ryan (2002) developed a model to construct robust crew schedules with bicriteria optimization. The authors' define a robust schedule to be one where disruptions in the schedule (due to delays) are less likely to be propagated into the future, causing delays of subsequent flights. Crew changing aircraft between operating sectors should occur less frequently in a robust schedule. The problem is formulated as a bicriteria problem, minimizing cost and non-robustness simultaneously. To solve the problem, the technique of minimizing only one objective, while transforming the others into constraints, specifying an upper bound on their values is used. The objective of minimizing cost is transformed into a constraint in this case and the transformed problem is solved using branch and bound.

### 2.3.2 Flexible Flight Schedules

The second broad classification of robust flight schedules are schedules with greater flexibility such that when a disruption occurs, recovery can be achieved with minimal alteration to the disrupted flight schedule. At present, only robust fleet assignment and robust aircraft rotation has been researched upon.

Ageeva (2000) defined robustness of a flight schedule as the extent of the flexibility that parts of the aircraft rotation schedule can be recovered in the event of irregularities in operations. A highly robust schedule may provide an option to reassign another aircraft to this routing and got back on its original routing before the next maintenance check. Robustness is measured by computing the percentage of points in the systems that have overlaps. *Point*, as defined by the author, is the interval of time that an aircraft spends at an airport between flights. Two aircrafts meet if they have points at the same airport within a same short interval. An *overlap* is an occurrence of two aircrafts meeting twice. Thus, the author’s robust fleet assignment model is one that included opportunities to swap planes.

Rosenberger et al. (2001b) also developed a fleet assignment model that can be used to improve robustness. It is based on the structure of a hub-and-spoke flight network to create a partial rotation with many short cycles. One of the major decisions that airline make to recover from an aircraft disruption is to cancel flight legs. By cancelling cycles, the rotation maintains flow balance without having to ferry an aircraft. The author’s approach of embedding many short cycles in the fleet assignment model is shown to perform better in operations. The robustness of such an assignment was demonstrated via a simulation of airline operations, SIMAIR. The on-time performance, percentage of legs cancelled, percentage of legs that are delayed on the runway or in the airspace for more than twenty-five minutes and number of legs that are flown by an aircraft different from the originally assigned one were used as measures of performance.

## 2.4 Evaluating robustness

Another interesting study related to schedule robustness is the models that are used to evaluate the robustness of a flight schedule.

In Haeme et al. (1998), the authors developed a Monte Carlo simulation model to help an airline evaluate its on-time arrival performance. The stochastic simulation of airline's operation allows the scheduler to test a variety of scheduling strategies and operations policies which might impact schedule performance. The model was built to represent the airline's entire hub-and-spoke operation. Using the model, the authors and airline operations planners were able to examine alternative strategies for maintaining high on-time performance without increasing costs. However, it was not known if approaches were developed to obtain a robust schedule.

Another simulation model, SIMAIR, was originally developed by Rosenberger *et al.* (2002). The idea was to develop a simulation tool to analyzed robustness of a prospective schedule and compare the effectiveness of different recovery strategies. Such a tool can potentially be very useful to schedulers as it would allow them to analyze the different performances of a prospective schedule. SIMAIR 2.0, an improved version of SIMAIR was later developed at NUS, by Lee et al. (2003). SIMAIR 2.0 is described in greater details in Chapter 5.

# 3 PROBLEM AND MODEL

From previous chapters, one would have realised that many studies have been conducted on airline schedule planning to optimize the decomposed sub-problems. These studies often aim to optimize the profits of the airline assuming that the schedule will be realised as planned. However, with frequent disruptions in airline operations, these schedules are far from optimal in practice. Robustness of the flight schedule thus becomes an issue of concern to the airlines. Little research has been carried out in this area to develop a more robust flight schedule. The researches that were studied previously did not consider the robustness of a flight schedule as an integrated problem, they mainly concentrate on constructing robust fleet assignment or crew scheduling independently and robustness is usually approximated by a mean value. Due to the complex interaction between these components, having a robust fleet assignment or a robust aircraft rotation only guarantees partial robustness but does not necessarily suggest an overall robust solution.

The motivation of this research is to investigate whether the overall performance of a flight schedule can be improved using an integrated approach, that considers the flight schedule problem as an entity, incorporating both the aircraft and crew. Improving the flight schedule in this study is accomplished by adjusting the departure times of each of the flights in an existing flight schedule. In this chapter, the problem is described in detail, and the mathematical formulation of the objective together with the constraints is presented.

## 3.1 Problem Description

The problem here can be described as follows: Given an existing flight schedule from an airline with its aircraft rotation and crew assignment determined, we seek to improve the robustness of the flight schedule by adjusting the departure times of the flights in the schedule.

In this study, we make the following assumptions.

**Assumption 1** An existing airline flight schedule and flight plan which includes the flight schedule, the aircraft rotation and the crew assignment is provided as an initial solution to the problem.

**Assumption 2** The aircraft rotation and the crew assignment is preserved, which means that aircrafts or crew, will not be rerouted.

**Assumption 3** The manner in which the airline operations is carried out is predetermined. i.e. the recovery policy used remains unchanged.

**Assumption 4** No flights will be cancelled and no additional flights will be created in the process.

## 3.2 Model Development

It is not unusual for airlines to desire to improve on more than one criterion when planning their flight schedule, by using conventional single-objective models, this would not be possible. Thus in this study, the problem is modelled as a multi-objective optimization problem; the following notations are defined prior to the mathematical model.

Decision Variables

$x_i$ Departure time of leg $i$

Parameters

$\tau_{B,i}$ Scheduled block time of leg $i$

$\tau_{Pconn}$ Minimum time for passengers to connect

$\tau_{Cconn}$ Minimum connection time for crew

$\tau_{Drest}$ Crew minimum rest after duty

$\tau_{BR}$ Duration of crew briefing before duty

$\tau_{DE}$ Duration of crew debriefing after duty

$T_D$ The latest time for the start of a duty where the crew will be granted additional rest.

$\tau_{Dmax,b}$ The maximum duration for duty starting before or at $T_D$

$\tau_{Dmax,a}$ The maximum duration for duty starting after $T_D$

$\tau_{Aturn}$ Minimum turn time for aircraft

$\tau_{M,i}$ Duration of scheduled maintenance after leg $i$

$T_{E,s}$ Earliest time a leg can depart from station $s$

$T_{L,s}$ Latest time a leg can arrive at station $s$

Indices

$l(d)$ Last leg of duty $d$

$f(d)$ First leg of duty $d$

$L$ Set of legs $i \in L$

$L_D(s)$ Set of legs departing from station $s$

$L_A(s)$ Set of legs arriving at station $s$

$D$ Set of duties $d \in D$

$D_b$ Set of duties that starts before or at $T_D$

$D_a$ Set of duties that starts after $T_D$

$S$ Set of stations $s \in S$

$C_P$ Set of passenger connections, where passengers connects from one legs to another in a passenger itinerary.

$C_{C1}$ Set of crew connections, where crew connects from one leg to another within the same duty

$C_{C2}$ Set of crew connections between two duties

$C_{A1}$ Set of aircraft connections without scheduled maintenance in between

$C_{A2}$ Set of aircraft connections with scheduled maintenance in between

The multi-objective problem can be formulated as

**Problem P1**

Objective Function

$$\min\left\langle f_1(\hat{x}), f_2(\hat{x}), \ldots, f_n(\hat{x}) \right\rangle$$

Subject to:

$$x_i + \tau_{B,i} + \tau_{Pconn} \le x_j \quad \forall (i,j) \in C_P \tag{2.1}$$

$$x_i + \tau_{B,i} + \tau_{Cconn} \le x_j \quad \forall (i,j) \in C_{C1} \tag{2.2}$$

$$x_i + \tau_{B,i} + \tau_{DE} + \tau_{Drest} + \tau_{BR} \le x_j \quad \forall (i,j) \in C_{C2} \tag{2.3}$$

$$x_{l(d)} + \tau_{B,l(d)} - x_{f(d)} + \tau_{BR} + \tau_{DE} \le \tau_{D\max,b} \quad \forall d \in D_b \tag{2.4}$$

$$x_{l(d)} + \tau_{B,l(d)} - x_{f(d)} + \tau_{BR} + \tau_{DE} \le \tau_{D\max,a} \quad \forall d \in D_a \tag{2.5}$$

$$x_{f(d)} - \tau_{BR} > T_D \quad \forall d \in D_a \tag{2.6}$$

$$x_i + \tau_{B,i} + \tau_{Aturn} \le x_j \quad \forall (i,j) \in C_{A1} \tag{2.7}$$

$$x_i + \tau_{B,i} + \tau_{M,i} \le x_j \quad \forall (i,j) \in C_{A2} \tag{2.8}$$

$$x_i \ge T_{E,s} \quad \forall i \in L_D(s) \text{ and } s \in S \tag{2.9}$$

$$x_i + \tau_{B,i} \le T_{L,s} \quad \forall i \in L_A(s) \text{ and } s \in S \tag{2.10}$$

The overall objective of the problem is to improve the robustness of the flight schedule; A flight schedule is considered robust if it is able to perform relatively well in various different situations. In other words, a schedule is robust if it is as insensitive to real life variabilities as possible (Mederer and Frank, 2002). Hence, the objective of the problem is minimizing several individual objectives, $f_i(\hat{x})$ simultaneously, where each individual objective is a measure of robustness e.g. percentage of flights cancelled. A list of the different measures that can be used to assess the robustness of a flight schedule was provided in Chapter 1. The decision variable $\hat{x}$ is a vector of decision variables, $\hat{x} = \{x_1, x_2, ..., x_Q\}$ such that each $x_i$ is a departure time of a flight leg in the flight schedule.

Constraint (2.1) ensures that passengers on an itinerary will be able to transfer to the next plane at the airport where transit is to be made. This is achieved by ensuring that the next leg in a passenger itinerary leaves later than $\tau_{Pconn}$ after the arrival time of the previous leg.

Constraints (2.2) and (2.3) ensure that crew can connect within and between duties respectively. Within the same duty, the crew needs a minimum connection time to be able to transfer from one aircraft to the next; constraint (2.2) makes sure that the departure time of the next leg in a crew duty is later than $\tau_{Cconn}$ after the arrival of the previous leg. From one duty to the next, the crew requires a minimum amount of rest. A duty consists of a set of legs flown by a crew in a day and the elapsed time of a duty includes a briefing period before the first leg of the duty and a debriefing after the last leg of the duty, so the actual rest time for the crew starts only after the debrief of the previous duty and ends before the briefing of the next duty. The Constraint (2.3) ensures that the departure time of the first leg in the next duty allows sufficient time for the crew to be debriefed for the previous duty, to rest and to be briefed for the next duty.

Constraints (2.4) and (2.5) ensure that the elapsed time, which is the duration of a duty, is within the permitted limits. Constraint (2.4) is for duties that start before $T_D$, and it ensures that the different components of the elapsed time added together does not exceed the limit. $T_D$ is a specific time, for instance 0600 hours, where crew that starts their duty before this time is only limited to fly a certain amount of time, and crew with duty starting after this time is allowed to fly for a slightly longer duration. Constraint (2.5) is a similar constraint to constraint (2.4) for duties that start after $T_D$. To simplify the problem, Constraint (2.6) is formulated to restrict duties starting after $T_D$ to its time interval such that if the departure time of the originally given flight schedule is after $T_D$, the adjusted departure time should also be after $T_D$.

Constraint (2.7) is to ensure that the aircraft will be able to turn to the next leg in its rotation in the case where there is no scheduled maintenance between the two legs.

This ensures that the next leg in the rotation of the aircraft leaves later than $\tau_{Aturn}$ after the arrival of the previous leg. For a rotation with a scheduled maintenance between the two legs, constraint (2.8) ensures that the next leg in the rotation of the aircraft leaves later than $\tau_{M,i}$ after the arrival of the previous leg.

Certain stations are not opened throughout the night; they have a limit on earliest time a flight can depart and the latest time a flight can arrive a station, constraints (2.9) and (2.10) takes care of the earliest departure and the latest arrival of each flight at a station respectively.

The optimal departure time, $x_i$ , which is of interest to us, given in terms of the originally planned scheduled departure time is given as follows

$$x_i = \overline{x_i} + \Delta_i$$

where

$\overline{x_i}$ is the scheduled departure time of the original schedule

$\Delta_i$ is the shift in the departure time of the improved schedule from the original schedule

Equivalently, *problem P1* can be represented by *problem P2* by replacing the original scheduled departure time with the above expression.

**Problem P2**

Objective function

$$\min \left\langle f_1(\hat{\Delta}), f_2(\hat{\Delta}), \ldots, f_n(\hat{\Delta}) \right\rangle$$

Subject to:

$$\bar{x}_i + \Delta_i + \tau_{B,i} + \tau_{Pconn} \le \bar{x}_j + \Delta_j \quad \forall (i,j) \in C_P$$

$$\bar{x}_i + \Delta_i + \tau_{B,i} + \tau_{Cconn} \le \bar{x}_j + \Delta_j \quad \forall (i,j) \in C_{C1}$$

$$\bar{x}_i + \Delta_i + \tau_{B,i} + \tau_{DE} + \tau_{Drest} + \tau_{BR} \le \bar{x}_j + \Delta_j \quad \forall (i,j) \in C_{C2}$$

$$\bar{x}_{l(d)} + \Delta_{l(d)} + \tau_{B,l(d)} - \bar{x}_{f(d)} - \Delta_{f(d)} + \tau_{BR} + \tau_{DE} \le \tau_{D\max,b} \quad \forall d \in D_b$$

$$\bar{x}_{l(d)} + \Delta_{l(d)} + \tau_{B,l(d)} - \bar{x}_{f(d)} - \Delta_{f(d)} + \tau_{BR} + \tau_{DE} \le \tau_{D\max,a} \quad \forall d \in D_a$$

$$\bar{x}_{f(d)} + \Delta_{f(d)} - \tau_{BR} > T_D \quad \forall d \in D_a$$

$$\bar{x}_i + \Delta_i + \tau_{B,i} + \tau_{Aturn} \le \bar{x}_j + \Delta_j \quad \forall (i,j) \in C_{A1}$$

$$\bar{x}_i + \Delta_i + \tau_{B,i} + \tau_{M,i} \le \bar{x}_j + \Delta_j \quad \forall (i,j) \in C_{A2}$$

$$\bar{x}_i + \Delta_i \ge T_{E,s} \quad \forall i \in L_D(s) \text{ and } s \in S$$

$$\bar{x}_i + \Delta_i + \tau_{B,i} \le T_{L,s} \quad \forall i \in L_A(s) \text{ and } s \in S$$

In *problem P2*, the objective of the problem is altered to minimizing several, $f_i(\hat{\Delta})$ simultaneously. The decision variable $\hat{\Delta}$ is a vector of decision variables, $\hat{\Delta} = \{\Delta_1, \Delta_2, ..., \Delta_Q\}$ such that each $\Delta_i$ is the shift in the departure time of the improved schedule from the original schedule.

The following chapters propose a solution procedure to Problem P2 solve the multi-objective optimization problem of improving the flight schedule by using multi-objective genetic algorithms (MOGA), which are the combinations of genetic algorithms (GA) and Pareto optimization.

# 4 SOLUTION APPROACH

In this chapter, we propose a procedure of applying the approach of genetic algorithms to find the non-dominated solutions in the Pareto front of the multi-objective optimization problem that was described. The different components of the genetic algorithms are detailed followed by a description of the overall procedure. Before the solution procedure is introduced, we introduce the fundamentals of multi-objective optimization along with an overview of genetic algorithms. A description of the advantages of applying genetic algorithms to multi-objective problems is also provided, together with its differences from classical search procedures.

## 4.1 Multi-objective Optimization

Most real-world engineering optimization problems are multi-objective in nature, they normally have several objectives that must be satisfied at the same time.

The multi-objective optimization problem is the problem of simultaneously optimizing the $n$ objectives. A multi-objective optimization problem in its general form can be written as:

$$\text{Minimize } \left\langle f_1(\hat{x}), f_2(\hat{x}), \ldots, f_n(\hat{x}) \right\rangle$$

$$\text{Subject to } g_j(\hat{x}) \le 0, \quad j = 1,2,\ldots,m$$

where $\hat{x} = \{x_1, x_2, \ldots, x_Q\}$ is the vector of decision variables, and $Q$ is the number of decision variables.

In a multi-objective optimization problem, multiple objective functions need to be optimized simultaneously. As such, the notion of "optimum" has to be re-defined in this context and instead of aiming to find a single solution, the objective is to produce a set of good compromises or from which the decision maker will select one. This is because in the case of multiple objectives there does not necessarily exist a solution that is best with respect to all objectives because of incommensurability and conflicting objectives. A solution may be best in one objective but worst in another. Therefore, there usually exist a set of solutions for the multi-objective case which cannot simply be compared with each other. Hence, for a multi-objective optimization problem, we seek a set of non-dominated, alternative solutions known as the *Pareto-optimal* set, as introduced by Pareto (1896). Pareto-optimal solutions are also known as efficient, non-dominated or non-inferior solutions; and the set of Pareto-optimal solution are also called the efficient frontier or trade-off surface.

Assuming a minimization problem, a given solution vector $\hat{u} = \{u_1, \ldots, u_Q\}$ is said to *dominate* another solution vector $\hat{v} = \{v_1, \ldots, v_Q\}$ if and only if $\hat{u}$ is better than $\hat{v}$ in at least one objective. Figure 4.1 shows an illustration of Pareto dominance for a two-objective problem, where both objectives are to be minimized.

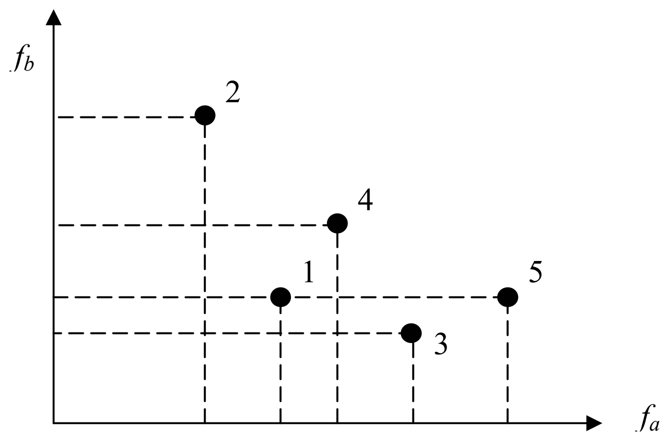



**Figure 4.1 A population of five solutions**

Five solutions with different objective function values are shown in the figure. If solutions 1 and 4 are compared, we observe that solution 1 is better than solution 4 in both objective values $f_a$ and $f_b$, hence, solution 1 dominates solution 4. Comparing another pair of solutions, solutions 1 and 5, solution 1 is better than solution 5 in objective value $f_a$ and equal in terms of objective value $f_b$. Since solution 1 is better in at least one objective, solution 1 is said to dominate solution 5. Considering yet another pair of solutions 1 and 2, solution 1 is better than solution 2 in objective value $f_b$, but solution 2 is better than solution 1 in objective value $f_a$. As the condition is not satisfied, solutions 1 and 2 do not dominate each other.

A solution vector $\hat{u}$ is said to be *Pareto-optimal* if there does not exist another solution vector $\hat{v}$ such that $f_i(\hat{u}) \le f_i(\hat{v})$ for all $i = 1,2,...n$ and $f_j(\hat{u}) < f_j(\hat{v})$ for at least one index *j*. Referring to the population of solutions in Figure 4.1, it is easy to see that solutions 1, 2 and 3 are not dominated by any other solutions, thus they are considered Pareto-optimal in this population.

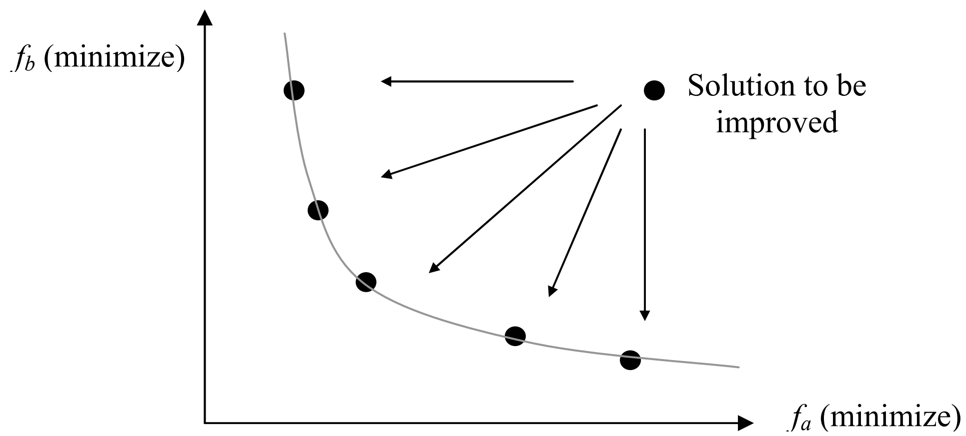



**Figure 4.2 Approaching the Pareto front for a two-objective problem**

Each point in Pareto-optimal set is optimal as no improvement can be achieved in one vector component that does not lead to degradation in at least one of the remaining components. Hence, none of the solutions in the non-dominated set is absolutely better than any other, any one of them is an acceptable solution. For a two-objective problem, to improve on the objective function values, the solutions will move in the general directions as indicated in Figure 4.2.

Over the years, a number of techniques were developed to deal with multi-objective optimization problems. However, it was only until recently, that researchers realized the potential of evolutionary algorithms in this area. The idea of applying evolutionary algorithms to multi-objective problems was first introduced by Rosenberg (1967), but this research area remained unexplored until recently.

## 4.2 Multi-objective Genetic Algorithms

### 4.2.1 Genetic Algorithms

Genetic algorithms (GAs) were formally introduced in the United States in the 1970s by John Holland at University of Michigan. Genetic Algorithms or Evolutionary algorithms are search and optimization procedures that are inspired by the principles of natural evolution.

The genetic algorithm maintains a population of individuals, where each individual or *chromosome* represents a potential solution to the problem. Each chromosome may contain a vector of decision variables, and each of these is known as the *gene*. GA is a method for moving from one population to another by using some form of natural selection with genetics-inspired operators. Each chromosome is evaluated according to some criteria, rating them in terms of their fitness. Some chromosomes are selected to undergo genetic operations of crossover and mutation which creates new chromosomes by combining parts from two chromosomes. The selection operator chooses those chromosomes in the population that will be allowed to reproduce, and on average, the fitter chromosomes produce more offspring than the less fit ones. After several generations, the algorithm converges to a solution which hopefully represents an optimal solution.

The steps involved for the "search for solution" using a GA are

- Generate a set of candidate solutions as initial solutions
- Evaluate the candidate solutions according to some fitness criteria
- Based on the fitness, select candidates (parents) that will undergo reproduction
- Produce further variants by using genetic operators on the selected candidates

There are two important issues with respect to search strategies: exploiting the best solution and exploring the search space. Through selecting the fitter chromosomes as parents to be reproduced, the genetic algorithms provide a directed random search in complex landscapes, thus exploiting the best solutions. Exploration of the search space is achieved through crossover and mutation, where genetic operators perform essentially a blind search toward the desirable area of the solution space.

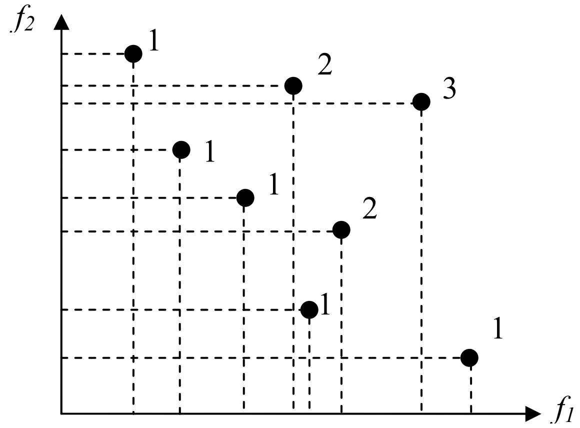
### 4.2.2 Multi-Objective Genetic Algorithms

During the last two decades, evolutionary approaches, in particular genetic algorithms have received considerable attention as an approach to solving multi-objective optimization problem. Classical search and optimization algorithms only use a single solution in each iteration; in contrast, genetic algorithms use a population of solutions. By involving a population of solutions in each iteration, the outcome of a genetic algorithm is also a population of solutions. If an optimization problem has a single optimum, genetic algorithm population members can be expected to converge toward the optimum solution. However, if an optimization problem has multiple optimal solutions, a GA can be used to capture multiple optimal solutions in its final population.

The basic feature of genetic algorithms is the multiple directional and global searches, in which a population of potential solutions is maintained from generation to generation. By dealing simultaneously with a population of possible solutions genetic algorithms allow us to find several members of the Pareto optimum set in a single run of the algorithm. They do not have any mathematical requirements and can handle all types of objective functions and constraints, thus, it can be used for optimization of theoretically known, as well as empirically estimated response surfaces. Also, genetic algorithms are less susceptible to the shape or continuity of the Pareto front (Coelle, 2001); they are able to handle very complicated structured multimodal response surfaces.

The first implementation of multi-objective genetic algorithms approach was Vector Evaluation Genetic Algorithm (VEGA) by Schaffer (1985). The VEGA algorithm differs from the simple GA in the way in which selection is performed. At each generation, a number of sub-populations are generated by performing proportional selection according to each objective function in turn. For a population with $k$ objectives and a population of size $M$, $k$ sub populations of size $M/k$ each will be generated. These sub-populations are then shuffled together to obtain a new population of size $M$, on which the GA will apply the crossover and mutation operators in the usual way.

Goldberg (1989) suggested the idea of Pareto ranking as a means of achieving equal reproductive potential for all Pareto individuals. These methods are based on the actual concept of Pareto optimality. The population is ranked on the basis of non-dominated individuals. The procedure is as follows. Assign Rank 1 to all non-dominated individuals and remove them from contention. Assign Rank 2 to the next set of non-dominated individuals and again remove them from contention. This process goes on until the entire population is ranked. This approach is shown in Figure 4.3, for a simple




case with two objectives to be minimized simultaneously.

**Figure 4.3 Pareto ranking of a population of solutions**

The rank-based fitness assignment proposed by Fonseca and Fleming (1993) explicitly caters to emphasize non-dominated solutions and simultaneously maintains diversity in the non-dominated solutions. This approach ranks the individuals according to the number of individuals in the current population by which it is dominated. All non-dominated individuals are assigned Rank 1, and any other individual, $\hat{\Delta}_i$, is assigned a rank equal to the number of solutions, $n_i$, that dominate solution $\hat{\Delta}_i$ plus one. Thus, for solution $\hat{\Delta}_i$, the rank assigned is

$$r_i = 1 + n_i$$

Note that not all ranks are necessarily represented in the population at a particular generation. This ranking method is illustrated in Figure 4.4.

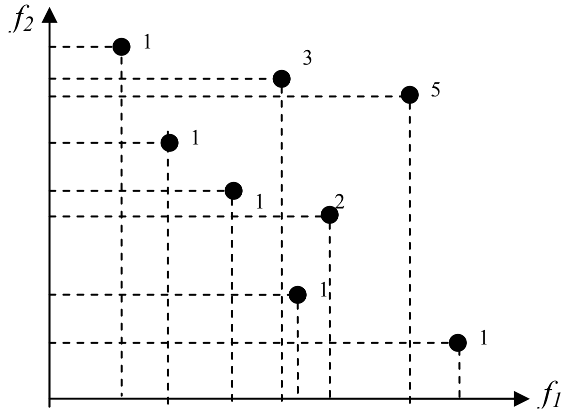



**Figure 4.4 Fonseca's method of ranking solutions of a multi-objective problem**

Srinivas and Deb (1994) developed the Non-dominated sorting genetic algorithm (NSGA). The population is ranked based on Goldberg's Pareto ranking method. The non-dominated individuals are first identified and assigned a large dummy fitness value which is proportional to the population size. To maintain diversity in the population, these individuals are shared (in decision variable space) with their dummy fitness values. After sharing, these non-dominated individuals are ignored temporarily and the second non-dominated front in the rest of the population is identified and assigned a dummy fitness value that is kept smaller than the minimum shared dummy fitness of the previous front. This process is continued until the entire population is classified into several fronts. A stochastic remainder proportionate selection is used to reproduce a new generation. Since individuals in the first front have the maximum fitness value, they always get more copies than the rest of the population. This allows us to search for non-dominated regions, and results in convergence of the population toward such regions. Sharing helps to distribute the population over the non-dominated region. Some researchers have reported that NSGA has a lower overall performance than MOGA, both computationally and in terms of quality of the Pareto fronts produced.

Horn et al. (1994) proposed the Niched Pareto Genetic Algorithm, which used a tournament selection scheme based on Pareto dominance. Two candidates for selection are picked a random from a population. A comparison set of individuals are also picked randomly from the population. Each candidate is then compared against each individual from the comparison set. Two types of out come may occur. First, one candidate is dominated by the comparison set and the other is not, in such a case, the non-dominated candidate is selected for reproduction. The second outcome is when both candidates are either non-dominated or dominated. Here, a sharing method is used to choose the winner for reproduction. In the sharing method used, the candidate with least niche count is selected as the winner. The niche count is calculated by counting the number of individuals in the population within a certain distance from the candidate. It was found that the performance of the NPGA method is sensitive several parameters. In particular the population size has to be large enough to search effectively and to sample the breadth if the Pareto front. It is also sensitive to the size of the comparison set which determines the selection pressure.

## 4.3 Components of the genetic algorithm

In this section, several basic components of the genetic algorithms procedure developed to solve the problem are described.

### 4.3.1 Coding Scheme

Each candidate solution (or chromosome), which represents a flight schedule, is represented as a real-number value vector. The coding scheme that is used to represent a chromosome is

$$\hat{\Delta} = \{\Delta_1, \Delta_2, ..., \Delta_Q\}$$

where $Q$ is the total number of flights in the entire flight schedule; $Q$ is equivalent to the number of decision variables in our problem. Each gene in a chromosome, $\Delta_q$ is a decision variable, corresponding to the shift in the departure time of the improved schedule from the original schedule.

### 4.3.2 Initialization

The genetic algorithm starts the search by generating a population of candidate solutions, each of which is a feasible flight schedule. Each solution is generated by perturbing the genes of the original solution by a small value. Each gene, $\Delta_q$, is a uniform random number generated within the interval [-$T_{range}$, $T_{range}$]. The candidate solution will then be

$$\hat{\Delta} = \{\Delta_1, \Delta_2, \ldots, \Delta_Q\}$$

### 4.3.3 Fitness Function and assignment

The evaluation of a solution in the multi-objective optimization problem is determining the objective function value for each of the objectives. Thereafter, a metric, known as the fitness, must be assigned to the solution based on all the objective values. The problem aims to optimize the robustness of the flight schedule, thus, it is intuitive that the performance measures, as listed in Chapter 1 will be used as a basis to assign the fitness.

#### 4.3.3.1 Evaluating the chromosomes

To evaluate a chromosome, which in this case is a flight schedule, we need to study its stochastic behaviour when the schedule is submitted to irregular disturbances. As it is, airline operation is an extremely involved process which is dependent on many different factors, most of which are uncontrollable. The departure time of a flight is not only dependent on the availability of the aircraft and crew; it is also highly dependent on external factors such as weather conditions and traffic situation at the airport. Due to the complexity of airline operations, testing the performance of a flight schedule by evaluating the objectives analytically is not viable; hence, simulation is the best approach to evaluate the performance matrix. In this study, SIMAIR 2.0, developed by Lee et al. (2003) is employed to evaluate each of the flight schedules suggested by the procedure. An elaboration of the simulation model used in SIMAIR 2.0 is given Chapter 5.

In this research, different flight schedules that are generated by our solution procedure will be tested by SIMAIR. SIMAIR will simulate the flight schedule and will generate the performance matrix, which will in turn be used as our objective function values for our multi-objective problem. Multiple replications by SIMAIR will be used to get an estimate of the performance of the flight schedule. Hence, while we are solving a planning problem of the airline schedule, the objectives are obtained from operations via simulation.

#### 4.3.3.2 Rank-Based Fitness Assignment

The rank-based fitness assignment proposed by Fonseca and Fleming (1993) is employed in this solution procedure. The method of ranking the population of solutions is as described in Chapter 4. After ranking the solutions, fitness values are assigned to each of the solutions. It is assigned such that a better fitness value is allocated to a solution in a better rank i.e. lower rank. Since the ranking procedure results in more than one solution in the same rank, the fitness values of the solutions in a rank is averaged out. This fitness averaging yields the following assignment of the average fitness to any solution $\hat{\Delta}_i$ where $i = 1, ..., N$ using the following equation:

$$F_i = N - \sum_{k=1}^{r_i - 1} \mu(k) - 0.5(\mu(r_i) - 1)$$

Where $r_i$ is the rank assigned to solution $\hat{\Delta}_i$ and $\mu(k)$ is the number of solutions in rank $k$.

This averaged fitness will be used for the parent selection procedure.

### 4.3.4 Parent Selection

The purpose of the parent selection in a genetic algorithm is to give more reproductive chances, on the whole, to those population members that are the most fit. By allowing the solutions that are fitter to be selected, the algorithm seeks to exploit the best solutions in the current population in hope that they will be able to produce offspring that are fitter. The parent selection procedure used here is the traditional Roulette wheel parent selection where each parent's chance of being selected is directly proportional to its fitness.

### 4.3.5 Crossover and Mutation

Crossover and mutation is performed after selecting the parents. Its function is to produce chromosomes created during the reproduction phase to create new solutions that deviate from those of their parents, hence exploring the search space.

#### 4.3.5.1 Crossover

Crossover is a process where the genetic material of two parents is recombined to form a pair of offspring. In this solution approach, two kinds of crossover are used simultaneously, the arithmetic crossover, and the one-point crossover.

*Arithmetic crossover* is commonly used for real coded solutions and in multi-objective optimization. In arithmetic crossover, random weighted mean of each gene between the two parents are computed to generate a new pair of solutions.

Let the parent solutions be

$$\hat{\Delta}_1 = \{\Delta_{1,1}, \Delta_{1,2}, \ldots, \Delta_{1,Q}\} \text{ and } \hat{\Delta}_2 = \{\Delta_{2,1}, \Delta_{2,2}, \ldots, \Delta_{2,Q}\}$$

Thus, the *q*th gene in *i*th solution is represented by $\Delta_{i,q}$.

Let the random weights be $x^{w1}$ and $x^{w2}$. Applying the first weight, $x^{w1}$, the newly created gene after arithmetic crossover is

$$\Delta'_{1,q} = (x^{w1} \times \Delta_{1,q}) + ((1 - x^{w1}) \times \Delta_{2,q})$$

For example, let the parents be

$$\hat{\Delta}_1 = \{3, 5, -4\} \text{ and } \hat{\Delta}_2 = \{-2, 4, 0\}$$

Let the random weights be $x^{w1}$ = 0.3 and $x^{w2}$ = 0.1. The pair of offspring after performing arithmetic crossover will then be

$$\hat{\Delta}_1' = \{0.3(3) + 0.7(-2), 0.3(5) + 0.7(4), 0.3(-4) + 0.7(0)\} = \{-0.5, 4.3, -1.2\}$$

$$\hat{\Delta}_2' = \{0.1(3) + 0.9(-2), 0.1(5) + 0.9(4), 0.1(-4) + 0.9(0)\} = \{-1.5, 4.1, -0.4\}$$

*One-point crossover* is a means of creating a pair of offspring by exchanging parts of the parents after a crossover point. The crossover point is a randomly selected integer, $x^{xoverpt}$ in the interval [1, *Q*]. Referring to the same pair of parents as before, the new pair of solutions after performing one-point crossover is

$$\hat{\Delta}_1' = \left\{\Delta_{1,1}, \Delta_{1,2}, \ldots, \Delta_{1,x^{xoverpt}}, \Delta_{2,x^{xoverpt}+1} \ldots, \Delta_{2,Q})\right\}$$

$$\hat{\Delta}_2' = \left\{\Delta_{2,1}, \Delta_{2,2}, \ldots, \Delta_{2,x^{xoverpt}}, \Delta_{1,x^{xoverpt}+1} \ldots, \Delta_{1,Q})\right\}$$

Let the crossover point be 2 for the arithmetic-crossover example, the pair of offspring that are produced is

$$\hat{\Delta}_1' = \{3, 4, 0\}$$

$$\hat{\Delta}_2' = \{-2, 5, -4\}$$

### 4.3.5.2 Mutation

Mutation maintains population diversity and is performed on each of the new solutions generated by the crossover procedure. It is randomly applied to each of the genes in a new solution. For each gene, $q$, a binary random variable, $x_q^{MUTATE}$, will be generated. These random variables will constitute the mask $x^{mask} = \{x_1^{MUTATE}, x_2^{MUTATE}, ..., x_Q^{MUTATE}\}$. If $x_q^{MUTATE}$ is 1, this gene undergoes mutation by generating another random variable $x_q^{mutate}$ in the interval [-$T_{mutate}$, $T_{mutate}$]. $T_{mutate}$ is a pre-assigned value in which we desire the gene to mutate in.

Let the parent be $\hat{\Delta} = \{\Delta_1, \Delta_2, ..., \Delta_Q\}$, after mutation, the $q$th gene, $\Delta_q$, would have transformed to

$$\Delta'_q = \Delta_q + x_q^{MUTATE} \times x_q^{mutate}$$

For instance, let a solution that will undergo mutation be

$$\hat{\Delta}_1 = \{3, 4, 0\}$$

and let the mask for $\hat{\Delta}_1'$ be $\{1, 0, 0\}$. This means that the first gene in $\hat{\Delta}_1'$ will be mutated. Let $x_1^{mutate} = -2$. The result of this mutation is a chromosome with values

$$\hat{\Delta}_1' = \{1, 4, 0\}$$

### 4.3.6 Formation of Child Population

After generating a full set of offspring, the parent population is combined with this set of offspring to form what we call a *combined population*. Thereafter, this combined population is ranked and the best will be placed in the child population, which will in turn be the parent population for the next generation. By selecting the best of the combined population, the algorithm ensures that the best member of the population will produce an offspring in the next generation.

### 4.3.7 Handling constraints and infeasible solutions

At every iteration, after each of the offspring is created, it is examined to check if any of the constraints is violated. If none of the constraints are violated, the solution is feasible and is automatically accepted. However, if one or more constraints are violated, the solution cannot be accepted directly. We do not discard the solution straight away; instead, we put forward a procedure that will bring infeasible solutions back into the feasible region.

Let the original solution (flight schedule) be $\boldsymbol{x}$, it should satisfy the set of constraints such that

$$A_j x \leq b_j \quad \text{or} \quad A_j x - b_j \leq 0 \qquad j = 0, 1, 2, \ldots, m \tag{4.1}$$

where $m$ is the number of constraints.

The amount of slack in constraint $j$ is $b_j$ - $A_jx$. Let the shift in the departure time of the new solution (generated by the GA) from the original schedule be **Δ**. For the new solution to be feasible, it must satisfy

$$A_j(x+\Delta)-b_j \le 0 \text{ or } (A_jx-b_j)+A_j\Delta \le 0 \tag{4.2}$$

A constraint $j$ is violated when

$$A_j\Delta \ge b_j - A_jx \tag{4.3}$$

For the solution to be feasible, the LHS of inequality (4.3) cannot be greater than $b_j - A_jx$. Thus, to bring the infeasible solution back into the feasible region, such that each of the constraints is satisfied, we divide the shift, **Δ**, by the change of the most violated constraint.

Let $j^*$ be the index of the most violated constraint such that

$$j^* = \arg\max_j \left[ \frac{A_j\Delta}{b_j - A_jx} \right]$$

The new solution (shift) after bringing it back into the feasible region is

$$\Delta' = \Delta \times \frac{b - A_{j^*}x}{A_{j^*}\Delta} \tag{4.4}$$

This new feasible solution will be accepted as a new offspring.

## 4.4 Overall procedure

The proposed algorithm stores two populations of solutions at every generation: an offspring population and an elite population. The *offspring population* contains the set of offspring that are generated in each generation, whilst the *elite population* contains the set of elite solutions in the each generation.

To make a fair comparison between two designs, replications will have to be performed by SIMAIR several times on both designs using the same set of random seeds. The set of random seeds is essentially the set of numbers that will be used by the random number generator of SIMAIR. In this way, for each simulation with the same random seed, different solutions are subjected to the same type of disruption conditions. In the solution procedure, one can set the number of offspring produced in each generation to a huge number, such as several hundreds or thousands. However, if that many offspring were produced in each generation, it will be too computationally expensive to assess the robustness of the design by using SIMAIR to evaluate each offspring through replications for different scenarios.

The purpose of having two populations of solutions is to reduce the computational time spent on evaluating each schedule. Let us denote the population size of the offspring population and the elite population as $N_{pop}$ and $N_{elite}$ respectively, where $N_{pop} >> N_{elite}$. We also denote $E_{pop}$ and $E_{elite}$ as the number of replications of simulation to estimate the performance of each offspring and each of the solutions in the elite population respectively, with $E_{pop} << E_{elite}$. At every generation, $N_{pop}$ offspring are generated from the parent population at the current generation. To keep the computation time down, SIMAIR evaluate these offspring with only $E_{pop}$ replications, that is, for a few disruption scenarios. Thereafter, $N_{elite}$ designs of the offspring population that perform best are extracted and place in the elite population. Each of the solutions in the elite population are then replicated in SIMAIR for $E_{elite}$ times (which should be a much larger number than $E_{pop}$) to estimate the performance of each flight schedule. By replicating each of the solutions in the elite population many more times using the same set of random seeds, each design is tested for a variety of disruption conditions. If the solution is able to perform relatively well in many different scenarios, this solution should be considered as a robust flight schedule. As GA is a stochastic search algorithm, the precise estimation of the design is not necessary. By the Pareto principle, only the best 15% of the design is significant to us. Thus, extracting the best solutions of the offspring to be replicated through iterations ensures that the good designs are not lost.

After the elite population of the current generation is assessed, it is combined with the elite population of the previous population and the combined best becomes the new elite population. In the same way, the offspring population is combined with the parent population and the best solutions are placed into the child population.

The algorithm can be terminated if the non-dominated chromosomes in the elite population remain unchanged for many consecutive populations.

Our algorithm can be written as follows.

**Step 1** Initialization: Generate an initial population (or parent population) of $N_{pop}$ solutions.

**Step 2** Evaluation: Use SIMAIR to estimate the performance of each solution in the parent population by replicating $E_{pop}$ times.

**Step 3** Fitness assignment: Assign a fitness value using the rank-based fitness assignment procedure to each of the solutions in the parent population.

**Step 4** Parent Selection: Select a pair of parents using the Roulette wheel parent selection method, this pair of parents will be used to generate a pair of offspring in the next step.

**Step 5** Crossover and mutation: Generate a binary random variable with probability PROB_XOVER to determine if crossover is to be applied to the parents. If crossover should be performed, determine the type of crossover to performed arithmetic crossover or one-point crossover with a 0.5 probability for each of them. This will result in a pair of offspring. For each of the solutions, apply mutation to the genes.

**Step 6** Feasibility check: For each of the solutions, check if it is feasible, i.e. of any of the constraints is violated. For infeasible solutions, apply the procedure to bring them back into the feasible region.

**Step 7** Check if the number of offspring is equal to $N_{pop}$. If there are insufficient number of offspring, go to step 4, else, proceed to step 8.

**Step 8** Determine the next elite population: Based on rank, select the best $N_{elite}$ solutions from the newly generated set of offspring, and let this set of solutions be the temporary elite population. Evaluate each of the designs in the temporary elite population using SIMAIR by replicating it $E_{elite}$ times. If it is the first generation of the algorithm, set the temporary elite population as the next elite population. Else, combine the previous elite population with the temporary elite population and select the best $N_{elite}$ to be the next elite population.

**Step 9** Formation of child population. The newly generated offspring and the old parent population are combined together and ranked. The best $N_{pop}$ is then selected. These solutions will make up the child population.

**Step 10** If the terminating condition is satisfied, end the algorithm. Otherwise, replace the parent population by the child population and the previous elite population by the next elite population for the next generation and return to step 2.

An illustration of the generation of the child and elite population at every generation is shown in Figure 4.5.

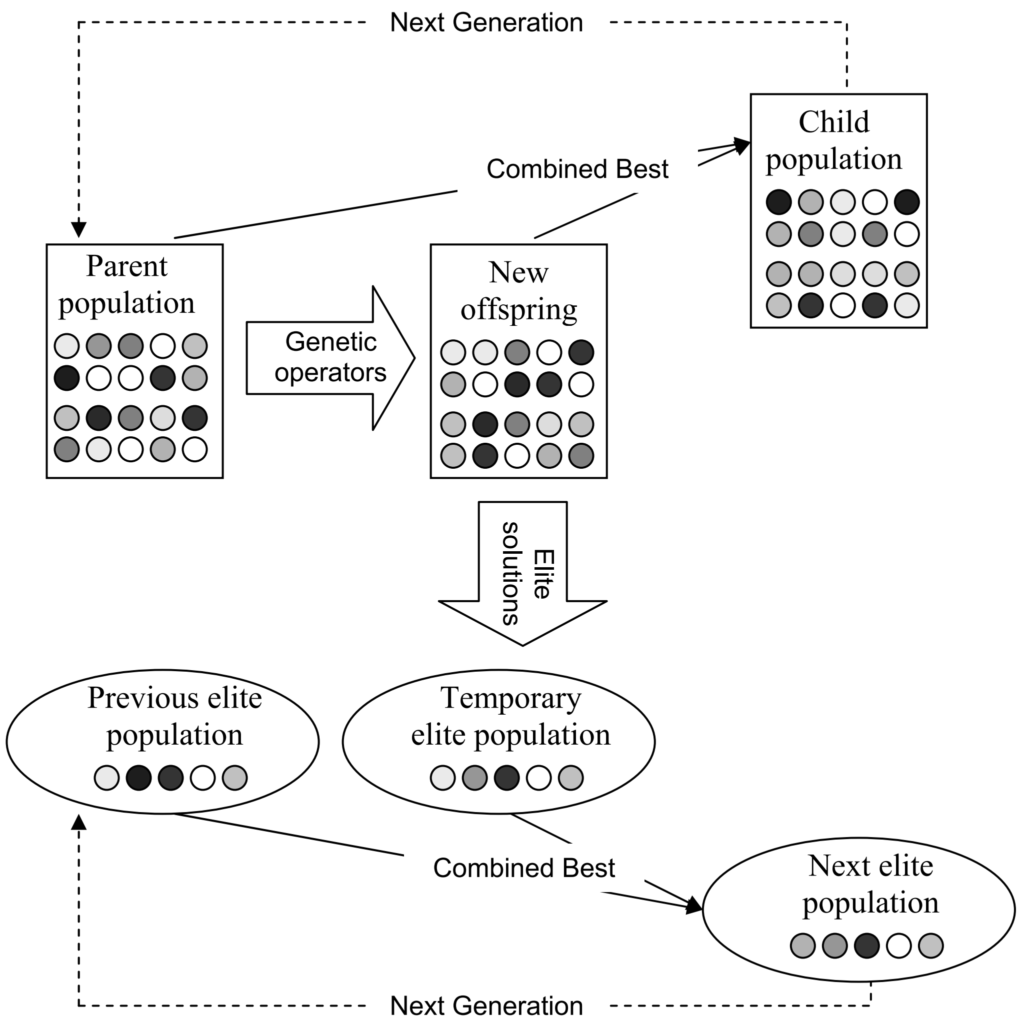



**Figure 4.5 Update of the child and elite population**

# 5 SIMULATION STUDY

The simulation model that is used to evaluate the robustness of flight schedules generated by the solution procedure is ‘SIMAIR 2.0’. SIMAIR 2.0 is a discrete simulation software developed by Lee et al. (2003), it is a C++ simulation tool that models the airline operations. SIMAIR 2.0 allows us to simulate the stochastic behaviour of airline operations when the flight schedule is subjected to disruptions. The output of the simulation is a set of performance matrix that can be used to assess the robustness of a flight schedule, which are essentially the objectives of our multi-objective problem.

A flight schedule is considered robust, as defined, if it is able to perform relatively well in various different situations of disruption. Different distributions for durations such as gate delay and probabilities such as probability of unscheduled maintenance are generated in SIMAIR by a random number generator using random seeds. By using different random seeds, SIMAIR simulates different disruption scenarios; each random seed corresponds to one disruption case. Hence, SIMAIR may return a different set of performance matrix each time the same design is evaluated with different random seed. When a flight schedule outperforms the others in a particular seed number, we cannot conclude that the schedule is robust as it is merely better in one scenario. Only when a solution is better than the rest of solutions in a variety of scenarios, can we infer that the solution is a superior solution. Thus SIMAIR allows us to replicate the simulation several times for one flight schedule, giving a good estimate of the performance of each solution.

## 5.1 Overview of SIMAIR 2.0

SIMAIR provides a means for devising and evaluating various airline recovery mechanisms to handle disruptions, and can also be used as a tool to evaluate the performance of a given schedule of operations.

A simplified overview of the operational SIMAIR model is shown in Figure 5.1. The inputs to SIMAIR are the flight schedule and the plan. The *flight schedule* is made up of legs that the airline will fly with its fleet and crew. The *flight plan* of an airline is the solution to the aircraft rotation and crew assignment problem.

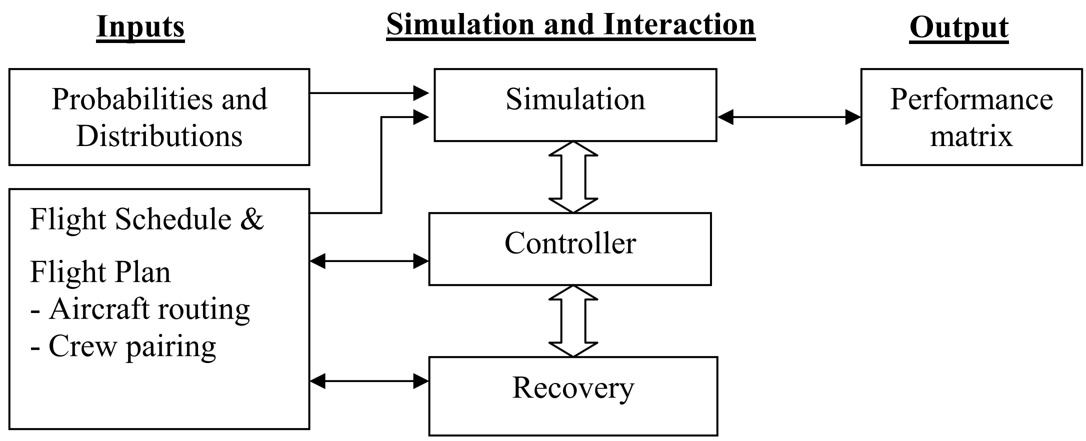



**Figure 5.1 An overview of the operational SIMAIR model**

The *simulation* module is solely involved in simulating operations. It is made up of a few components, such as a future event list, an event scheduler, and a simulation clock. The stochastic aspects of the simulation like gate delays, and unscheduled maintenance are handled using a probability distribution.

The *controller* module plays the role of schedule legality checker. The controller is called at different stages of the simulation to check for schedule legality, ensuring that the aircraft and crew can fly. If legality is violated, this module will call upon the recovery to suggest a legal flight schedule.

*Recovery policies* are different methods or policies airlines use to deal with disruptions or irregularities. After the recovery module comes up with an alternate plan, the controller module will first check for legality of the proposed plan, and then implement the changes recommended if the proposed alternative is legal.

The *performance matrix* is a matrix consisting of the different measures that can be used to quantify the performance of a flight schedule. A list of some of the performance measure that is computed by SIMAIR is given in the next section.

SIMAIR is conceptualized and organized in a modular way that allows, as much as possible, the ease of integration of recovery modules written by different researchers or airlines. It also allows for inclusion of different crew and aircraft legality rules, making it easier to customize SIMAIR for simulating the operations of any specific airline with specific fleet and crew requirements.

### 5.1.1 Simulation module

The simulation module models the plane's operation as a sequence of events. One event triggers another leading to a simulation of airline operations. Each leg in the schedule can be decomposed according to seven events, which are determined by the queuing network as shown in Figure 5.2. Details of each event are as follows.

- **Scheduled Departure Event**. Pilot and passenger scheduled to depart from the gate.
- **Depart Gate Event**. Plane pushes away from the gate and begins to taxi to the runway.
- **Enter Runway Queue Event**. Plane enters the runway queue of the departure station.

- **Leave Ground Event**. Plane reaches the front of the runway queue and begins its takeoff.

- **Arrive Airspace Event**. Plane arrives at the airspace of the arrival station and enters the airspace queue at that station.

- **Touch Down Event**. Plane reaches the front of the airspace queue and begins to land.

- **Arrive Gate Event**. Plane reaches the gate at the arrival station.

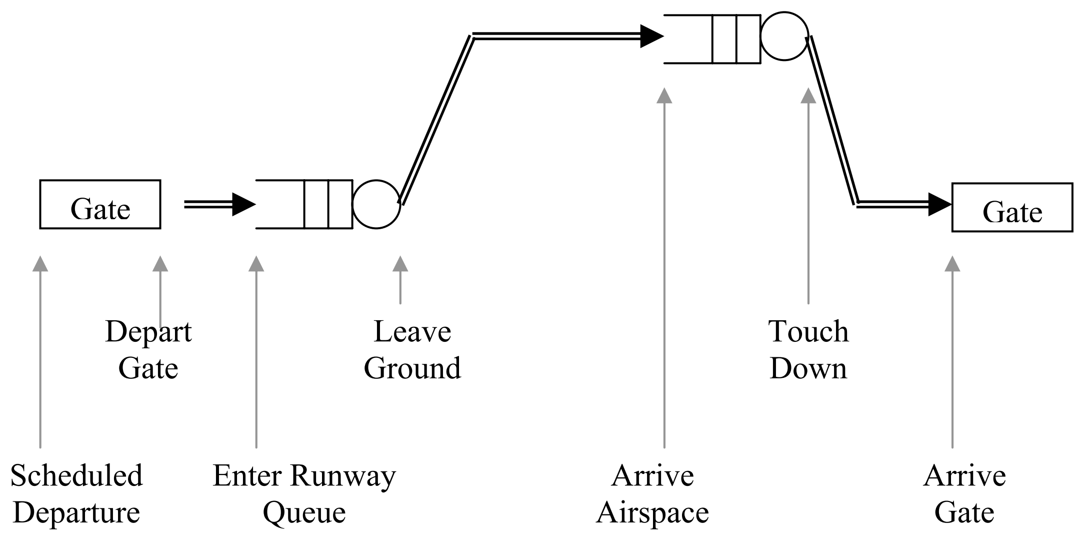



**Figure 5.2 Decomposition of a leg**

In addition to the seven events which make up the components of a leg, there are five additional events.

- **Enter Major Unscheduled Maintenance Event**. Plane is required to undergo major unscheduled maintenance. It is a chance event that is generated after the depart gate event.

- **Enter Minor Unscheduled Maintenance Event**. Same as the Enter major unscheduled maintenance event, except that the duration generated for the unscheduled maintenance is much shorter.

- **Leave Major Unscheduled Maintenance E**vent. This event is generated after plane undergoes major unscheduled maintenance, and signals the simulation module that the plane is now ready to fly again.

- **Leave Minor Unscheduled Maintenance Event**. Similar to the Leave major unscheduled maintenance, except that it is generated after the plane undergoes minor unscheduled maintenance.

- **Service Rate Event**. This is an event that changes the service rate of runways of airports simulated. This event changes the duration that a plane needs to take off and land. When the service rate is increases, this means that more planes can take off or land with a same time interval. In case of service rate event dropping to zero, the airport is closed and no planes can take off or land. The current version of SIMAIR does 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 modelled as the service rate of the airport.

At each station, planes are modelled to fly-in and fly-out as a first-in-first-out queue. To simulate this action, a runway queue and airspace queue are modelled. The runway queue is for the aircraft beginning their flight that need the runway for takeoff and the airspace queue is for aircraft that will need the runway to land. These queues are sequences of airplanes that are served at a rate equivalent to that of the service rate of the airport, which in-turn depends on weather and congestion at that particular airport. The queues are assumed to have infinite capacity.

SIMAIR generates random variables for ground time (duration of time between Depart Gate Event and Enter Runway Queue Event), block time (duration of time between Leave Ground Event and Arrive Airspace Event) and unscheduled maintenance duration.

### 5.1.2 Controller Module

The simulation module, in the course of execution, calls the controller module at the beginning of every event. The controller module accounts for rules and regulations enforced by bodies like the FAA by introducing the concept of legality. Airline operations are often governed by mandatory rules, such as those proposed by the FAA and those agreed upon by crew unions, regarding the deployment of planes and crew respectively, in operations. Besides regulations imposed upon the airline, there are also many other constraints that have to be met before an aircraft can be considered legal to takeoff. The controller also checks for the violations of these constraints. One such constraint is to ensure that the aircraft that will be used to fly the leg is available at the origin airport.

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. For example, at schedule departure event, one might want to check aircraft legality, for example, whether the crew flying the leg is available.

Two types of illegality are identified by the controller module: Immediate illegality and future illegality. Immediate illegality will render the next leg infeasible while future illegality will only cause a problem some time in the future, if the simulation continues as it is.

In the event of illegalities, the controller module will call the recovery module to fix the problem. The controller module will pass the necessary information, such as the type of illegality encountered, the plane or crew involved in the illegality, etc., to the recovery module to allow it to come up with a new flight schedule that is legal.

The controller module is also responsible for checking the feasibility of the proposed changes to the schedule recommended by the recovery module. Some recovery policies might choose to ignore future illegalities passed from the controller module, and only fix the problem for immediate illegalities. Other recovery policies might prefer a "proactive" approach and fix the illegality as soon as it appears (without hindering the immediate execution of the simulation). Since SIMAIR is to be used by different recovery policies, provisions are made such that the controller module will only make sure that the proposed changes ensure that the immediate next leg is legal. If it is not, the controller module will call the recovery module again. The controller module will not call the recovery module again if only future illegality is encountered.

Once the proposed changes by the recovery module are accepted, the controller module will have the additional role of implementing the changes to the operational schedule.

**Table 5-1 Parameters used in the 8-in24 hours rule**

| Scheduled Aloft within 24 hours | Hours of Scheduled Rest | Minimum hours of reduced rest | Hours of compensatory rest |
|---|---|---|---|
| $< 8$ hours | $\geq 9.5$ | 8 | 10 |
| $8 <$ blockTime $< 9$ | $\geq 10$ | 8 | 11 |
| $> 9$ hours | $\geq 11$ | 9 | 12 |

The operational crew legality rules that are checked by controller module include the 8-in-24 hours rule, the 30 hours in 7 days rule and the maximum duty duration rule.

The 8-in-24 hours rule is a rule that restricts the airline from making any crew fly for more than 8 hours in any 24 hour time window. However, this rule can be relaxed in operations the crew can fly for a longer duration provided that he is given extra rest at the end of the duty. The parameters used in the 8-in-24 hours rule are given in Table 5-1. The *schedule aloft* is the actual flying time of a crew in a duty, the *hours of scheduled rest* is the amount of rest the crew is originally entitled to, the *minimum hours of reduced rest* is the minimum number of hours the duration of rest given to the crew can be reduced to and the *hours of compensatory rest* is the duration of rest that has to be given to the crew if his previous rest was reduced to a reduced rest.

The 30 hours in 7 days rule is a much simpler rule that is similar to the 8-in24 rule. It basically limits the flying time of any crew member to a maximum of 30 hours for a window of 7 calendar days.

Maximum duty duration is enforced on every duty that a crew performs. If the crew starts his duty before 0600 hours, he can fly up to a maximum of 10 hours, while if he starts on or after 0600 hours, he can fly up to a maximum of 12.5 hours.

### 5.1.3 Recovery Module

A general framework for the recovery module has been established in SIMAIR. Currently, a simple default recovery policy is in place, but users can substitute their recovery policies by following the general framework.

The default recovery policy in place 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:

- Use of reserve crew in event regular crew unable to fly the next leg.
- Deadheading of regular crew to crew bases.
- Pushback of flights when the delay is lower than a threshold and still maintains schedule feasibility.
- Cancellation of several flights or 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 i.e. the airport reopens.
- Ferrying of aircraft to stations with maintenance capability to ensure maintenance feasibility.

Conceivably, users of SIMAIR can use some other options to recover, notably utilization of spare aircraft at certain airports, or aircraft swapping. These recovery actions can be coded into SIMAIR.

### 5.1.4 Performance Measures

A series of performance metrics used in evaluating the schedule have been coded into SIMAIR. The output data that are collected at the end of simulation includes a set of summarized data and raw data on each of the stations, flight legs, crew and group of passengers.

For each station in the system, SIMAIR keeps track of the number of flights that are on-time and the number of cancelled flights.

Summarized data for flight legs include information such as the number of legs flown, number of legs cancelled and number of legs that are on-time. Also provided, is the frequency of legs that are late for less than fifteen minutes, between fifteen and thirty minutes etc.

Summarized data for crew on the other hand, contains information on the frequency that different crew legality rules are violated, the number of times reserve crew are called upon to replace a regular crew and the number of times crew are deadheaded. Also recorded is the operational crew costs (FTC) of the crew.

Passengers that fly on the same series of flights from an origin to a destination are said to be on the same passenger itinerary. In the summary of passenger data, the number of passengers that missed their connections, number of passengers that are inconvenience, the number of itineraries that are disrupted and the lateness of passengers are kept track off.

SIMAIR also keeps a record of all the raw data. The raw data contains the unprocessed information about each leg categorized into under different classes, by the airports that are utilized, the aircraft and crew that are used to fly the leg and by the passenger itinerary that contains the legs. These information consist of the details of each leg that has flown, the time of the occurrence of each of the events such as the scheduled departure event and unscheduled maintenance event. The raw data are collected to allow users to trace the various events that had happened to each leg. The provision of these raw data gives the user the flexibility to process the data into statistics that are meaningful to them.

## 5.2 Measure of Robustness

In this study, we optimized two objectives of the robustness is considered; the operational FTC and the percentage of flight delayed. SIMAIR is employed to evaluate these two measures of robustness for each of the flight schedules generated by the solution procedure. The proposed solution approach can however be extended to optimize more than two objectives simultaneously.

### 5.2.1 Operational FTC

The operational FTC (flight-time-credit) is used by most airlines in United States to assess the operational cost of a crew schedule.  FTC is the difference between the number of minutes paid and the number of minutes flown, as a percentage of the number of minutes flown, defined as

$$\mathrm{FTC} = \frac{\text{(pay - and - credit minutes) - flytime}}{\text{flytime}} \times 100\%$$

The number of pay-and-credit minutes that an individual crew accumulates is referred to as the crew cost. For each leg *f*, let originalblock(*f*) be the originally planned block time leg *f*. Let elapsed be the planned elapsed time of duty *d*. Let $r_e$ be a fraction representing the rate of pay for the elapsed time, and let *mgd* be the minimum guarantee for a duty. The planned duty cost of duty *d* is assumed to be

$$b_d = \max\left\{\sum_{f\in d} \text{original block}(f), r_e \times \text{elapse}_d, mgd\right\}$$

Let TAFB$p$ be the planned time away from base of pairing $p$. Let $r_t$ be a fraction representing the rate of pay of *TAFB*. Let *mgp* be a minimum guarantee per duty in a pairing, and let numduties$_p$ be the number of duties in pairing $p$. The planned pairing cost of pairing $p$ is

$$c_p = \max\left\{\sum_{d\in p} b_d, r_t \times TAFB_p, mgp \times numduties_p\right\}$$

Vance et al. (1997) use values of $r_e = \frac{4}{7}, mgd = 0, r_t = \frac{2}{7}$, and $mgp = 300$.

One of the methods of computing the actual pairing cost is described here.

Operational duty cost of a duty $d$, is given by

$$b_d' = \max\left\{\sum_{f\in d} \text{block}(f), r_e \times \text{elapse}_d, mgd\right\}$$

where block($f$) is the operational block time of leg $f$, and elapse$_d$ is the operational elapsed time of duty $d$.

Operational pairing cost of pairing $p$ is given by

$$c_p' = \max\left\{\sum_{d\in p} b_d, r_t \times TAFB_p, mgp \times numduties_p\right\}$$

where *TAFBp* is the operational time away from base of pairing $p$.

The actual pairing cost of the pairing $p$ is

$$\tilde{c}_p = \max\{c_p, c_p'\}$$

Let $P$ be the set of all pairings in a crew schedule, the actual number of pay-and-credit minutes is then given by

$$\sum_{p \in P} \tilde{c}_p$$

### 5.2.2 Operational Percentage of Flights Delayed

The percentage of flights delayed serves as a measure of timeliness of flights. A flight is considered late or delayed if it arrives after fifteen minutes of it scheduled arrival time. Cancelled legs are also considered late. This percentage is given by

$$\%\,\text{delayed} = \frac{\text{flights arrving 15 minutes after scheduled time + cancelled}}{\text{total number of scheduled flights}}$$

## 5.3 Test Data

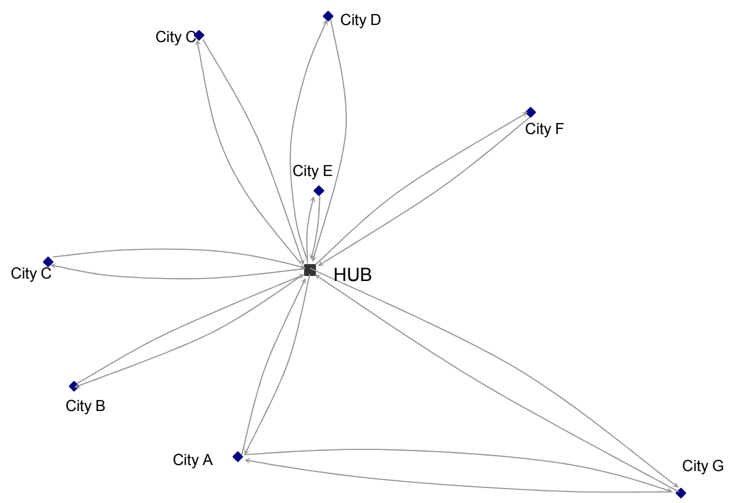



**Figure 5.3 Graphical representation of flight network used in test data**

In this thesis, we generated three test data sets (Test Data $A$, $B$ and $C$) for evaluating the performance of our solution procedure. Each test data set consists of three components, the flight schedule, aircraft rotation and crew pairing. The test data sets have the same flight schedule and aircraft rotation and differ in the crew pairing and parameters used. Here, we provide a brief description of how our test data sets were generated.

### 5.3.1 Generating the Flight Schedule and Aircraft Rotation

A typical American airline has several fleet types and flies several hundreds of domestic legs everyday using these fleets. In this thesis, the flight schedule that was used as our test data sets is a subset of the flight schedule which was extracted from one of the American Airlines. The original flight network had a hub-and-spoke network structure serving many cities around US; most of the flight legs fly in and out of the hub. To arrive at the flight schedule used as our test data, we selected the flight legs that were flown by one particular fleet type to be included in our flight schedule. A graphical representation of the approximate geographical location of the airports selected for our test data sets is shown in Figure 5.3. Most of the flights fly in and out of the hub, only several flights in a day are point to point flights that do not pass through the hub. The airline flies the same legs everyday of the week, thus, our schedule is a week long flight schedule. 63 flights are scheduled each day, making up a total of 441 flights in seven days.

### 5.3.2 Generating the Crew Schedule

The difference between the three sets of test data lies in the crew pairing structure. These crew pairings were constructed manually, observing all the legality rules of the crew.

**Test Data *A***

The characteristics of this set of crew pairings is such that each duty consists of an average of two flight legs. Since the flight network has a hub-and-spoke structure, we assign a crew to two consecutive legs that fly in and out of the hub. Figure 5.4 shows examples of how the crew is assigned to a flight. Leg 1 departs city A at 0620 hrs and arrives at HUB at 0735hrs while leg 25 departs HUB at 0855hrs, arriving city A at 1000hrs. The crew legality rules (e.g. crew connecting time at the HUB within permissible limits and total flying time within limits) were checked to be non-violating, thus leg 1 and leg 25 were assigned to the same crew. Another example is assigning flight legs 23 and 10 to the same crew.

By assigning crew manually using the above method for our week long flight schedule, 126 crew members including pilots and first officers were scheduled to fly 63 pairings, consisting of a total of 203 duties.

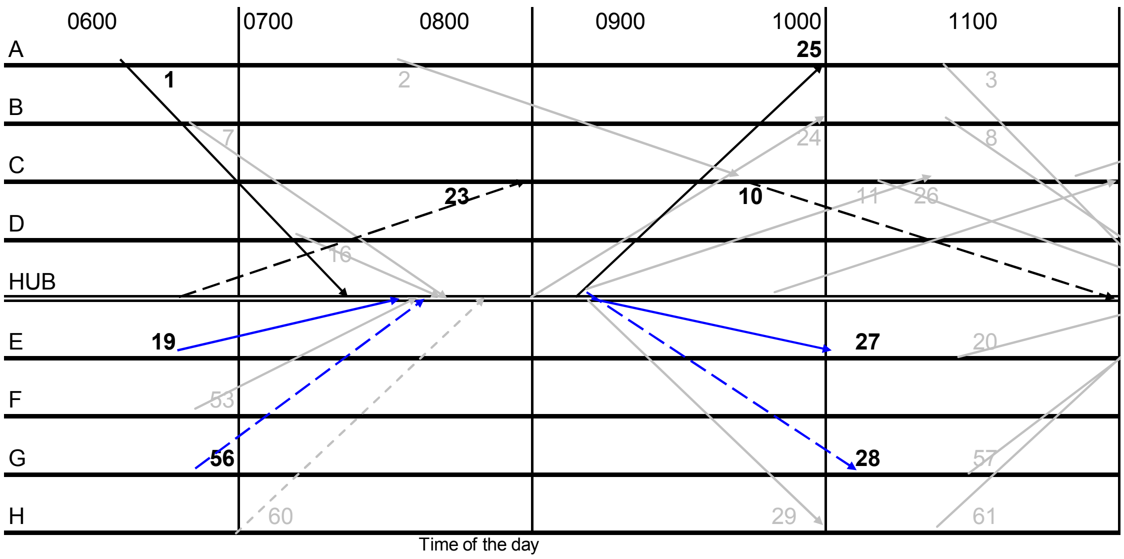



**Figure 5.4 Time representation of flight schedule used in test data *A***

The input files for this set of test data can be found in Appendix A.

**Table 5-2 Crew structure for test data sets**

| | Test Data *A* | Test Data *B* |
|---|---|---|
| No. of Crew members | 126 | 110 |
| No. of Pairings | 63 | 55 |
| No. of Duties | 203 | 161 |

**Test Data *B***

In test data *B*, instead of flying an average of two legs a day, most of the crew flew three legs in a day and the pairings lasted three or four days. Figure 5.5 shows the timeline representation of the flight schedule used in test data *B* and the crew assigned to the flight legs. Legs 1, 25 and 3 are assigned to the same crew while legs 19, 27 and 20 are assigned to another crew. With each crew flying more legs each day, fewer crew members were needed. The number of crew members, number of pairings and number of duties are compared against the original test data in **Error! Not a valid bookmark self-reference.**. Crew duties and pairings were created in such a way, to investigate how FTC is affected by the structure of crew duties and if more improvements can be achieved in terms of the non-dominated solutions.

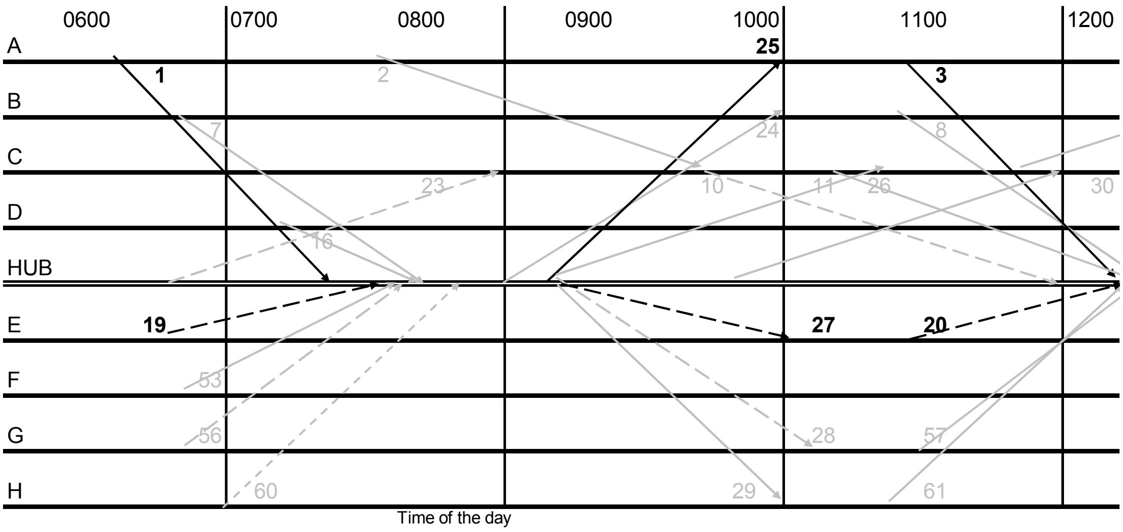



**Figure 5.5 Time representation of flight schedule used in test data *B***

**Test Data *C***

Test Data *C* was generated after the results for test data *A* has been obtained. They have the same flight schedule and aircraft rotation, and the difference is parameters used in the computation of FTC. The motivation behind adjusting the parameters of FTC is to investigate if a change in these parameters would result in different values for the final computed FTC and whether this would lead to better solutions in terms of the two objectives that were optimized. A comparison between the parameters used is given in Table 5-3.

**Table 5-3 Set of Parameters used to compute FTC**

| | Test Data *A* | Test Data *C* |
|---|---|---|
| $r_e$ (fraction representing the rate of pay for the elapsed time) | 4/7 | 5/9 |
| $r_t$(fraction representing the rate of pay of *TAFB*) | 2/7 | 2/7 |
| *mgd*(minimum guarantee for a duty) | 0 | 250 |
| *mgp* (minimum guarantee per duty in a pairing) | 300 | 325 |

The adjustment in parameter used to compute FTC is based on the results that were obtained for Test Data *A* (refer to Chapter 6). In the results for Test Data A, pairing cost for the extracted pairing is

$$\begin{aligned} c_p &= \max\left\{\sum_{d \in p} b_d, r_t \times TAFB_p, mgp \times numduties_p\right\} \\ &= \max\left\{1279, \frac{4}{7} \times 5484, 1200\right\} \end{aligned}$$

It was also noted that for most of the pairings, *TAFB* had the highest value. Thus to make the last component comparable, *mgd* was raised from 300 to 325.

Duty costs for each of duty in the pairing can be computed by

$$b_d = \max\left\{\sum_{f \in d} \text{original block}(f), r_e \times \text{elapse}_d, mgd\right\}$$

For each duty in pairing 1907, the values of each component are displayed in Table 5-4. Similarly, the second component, which is dependent on the elapsed time of each duty has the highest value; this is also true for most other duties in other pairings. In Test Data *C*, $r_e$ is lowered from 4/7 to 5/9 and *mgd* is given a value of 250.

**Table 5-4 Computation of each duty cost in pairing 1907 for Test Data *A***

| | original block($f$) | $elapse_d$ | $r_e \times elapse_d$ | $mgd$ | $b_d$ |
|---|---|---|---|---|---|
| Duty 1 | 238 | *693* | 396 | 0 | **396** |
| Duty 2 | 165 | *520* | 297 | 0 | **297** |
| Duty 3 | 168 | *514* | 294 | 0 | **294** |
| Duty 4 | 161 | *512* | 293 | 0 | **293** |

By changing the values of the parameters used to compute FTC. In this way, the values of the three components for duty cost and pairing cost become more competitive.

## 5.4 Parameter Setting

**Table 5-5 Values of Parameters used in solution procedure**

| Symbol | Description | Value |
|---|---|---|
| $N_{pop}$ | Number of solutions in the child population | 1000 |
| $N_{elite}$ | Number of solutions in the elite population | 20 |
| $E_{pop}$ | Number of times each design in the child population is evaluated | 1 |
| $E_{elite}$ | Number of times each design in the elite population is evaluated | 20 |
| PROB_XOVER | Probability of crossover | 0.6 |
| PROB_MUTATE | Probability of mutation | 0.2 |
| $T_{range}$ | The maximum range which each gene of the initial population solution is perturbed | 5 minutes |
| $T_{mutate}$ | The maximum range which each gene is mutated | 3 minutes |

The values of the parameters used for the solution procedure is shown in Table 5-5. For the solution procedure to be able to generate sufficient offspring at each generation, a value of 1000 was used $N_{pop}$. Applying the Pareto principle, only the best 15% of the design is significant to us, thus is would be meaningful to select 150 of the best solutions to be placed in the elite population. Nonetheless, due to time constraint, only the best 20 designs were selected for the elite population ($N_{elite}$). Each offspring generated by the procedure is evaluated by SIMAIR once ($E_{pop}$), without replicating, and the best designs in the elite population are replicated 20 times ($E_{elite}$) by SIMAIR to estimate the performance of the schedule.

For the initial population, each gene (flight departure time) is perturbed up to a maximum of 5 minutes earlier or later than the original departure time. At each generation, after a pair of parents is selected, they will undergo a crossover with a probability of 0.6. After crossover, mutation is performed on each flight departure time of the offspring with a probability of 0.2, if mutation is carried out; the departure time (gene) can be mutated up to 3 minutes earlier or later then its current departure time.

# 6 RESULTS

In order to assess the performance of the proposed approach to solve the described problem, the approach was tested on the Test Data *A* and Test Data *B* that were described in the previous chapter.

## 6.1 Test Data *A*

In the implementation of the solution procedure on the test data, the terminating condition was carried out by visual inspection, such that if little or no improved is made from one elite population to the next, the Pareto front has been achieved.

For the set of test data, the procedure was performed for 300 generations where improvements on the non-dominated front towards the $300^{th}$ generation front were nominal. Figure 6.1 illustrates the progression of the non-dominated front of the elite population from the initial stage up to the $300^{th}$ generation. The horizontal axis and vertical axis on the plots correspond to the values of the operational FTC and the operational percentage of flights delayed respectively. The labels of the axes are deliberately left out on the diagrams to avoid clutter.

Since the initial set of elite solutions were basically random perturbations of the original schedule, it is not surprising that some of the solutions were inferior to the original solution. However, as the genetic algorithm progress in search of better solutions it can be seen that the original schedule is dominated by more and more solutions.

It is also noticeable that the set of elite solutions moves in a general direction attempting to minimize both of the specified objectives, that is, the solutions moves downwards and leftwards through the generations. As such, the non-dominated front also shifts in the same general direction. The solutions on the non-dominated front are those that are of interest to the airlines, these solutions can be implemented to achieve better performance in operations. The non-dominated front of our final population will be examined in greater detail in the next section.

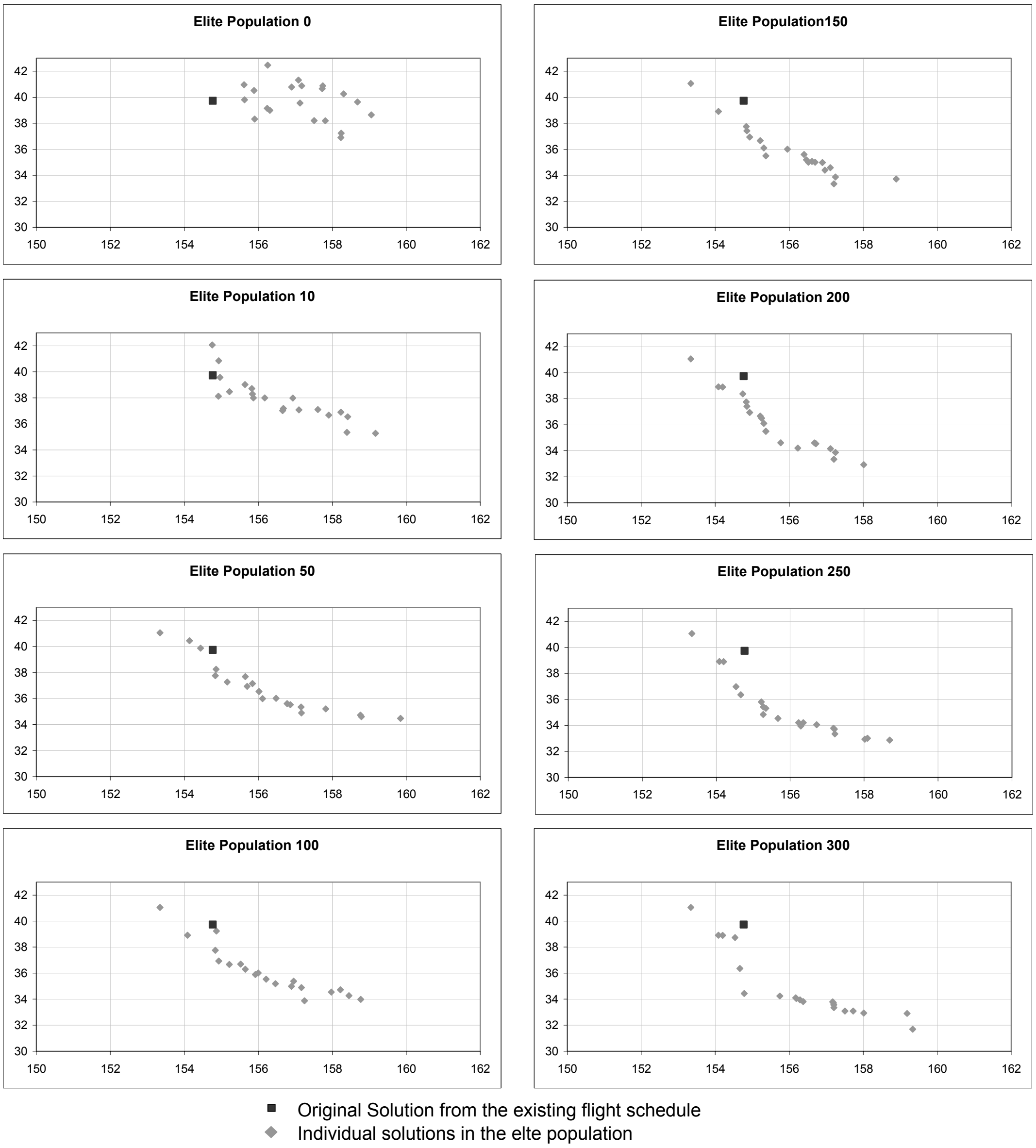



**Figure 6.1 Movement of elite population towards the Pareto front over several generations of the Genetic Algorithm**

To demonstrate that the non-dominated front has been achieved for the given set of flight schedule at the 300$^{th}$ generation, the solution procedure was further implemented for several hundreds of generations, with the elite population of the 300$^{th}$, 500$^{th}$ and

700th generation displayed in Figure 6.2. The solution procedure works such that the best offspring that are generated in each generation is re-evaluated with different disruption conditions (refer to section 4.4). These offspring are then compared with those that are already in the elite population, only the best $N_{elite}$ are kept in the elite population. In Figure 6.2, from the 300th generation to the 500th generation and then to the 700th generation, only very few (circled in the 300th generation plot) of the solutions in the elite population have been outperformed by new offspring generated by the parent population.

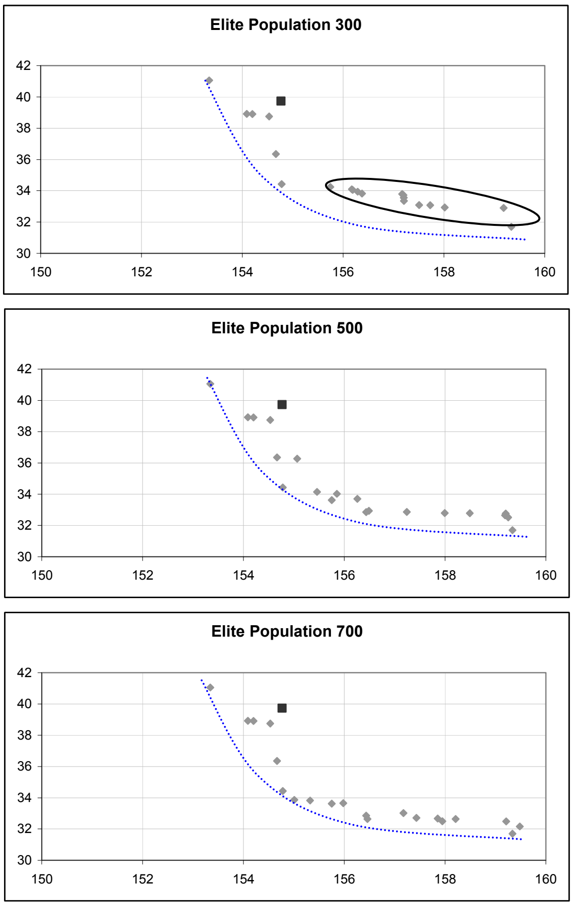



**Figure 6.2 Elite Population of generations 300, 500 and 700 of the Genetic Algorithm (for test data A)**

### 6.1.1 Non-dominated front

An idea solution that is optimal with respect to all objectives in general does not exist for a multi-objective problem. Trade-offs must often be made between different objective functions. By allowing the optimization of multi-objective, the airline is present with the set of non-dominated solutions, which forms the trade-off surface or what is known as the efficient frontier. The trade-off surface allows the airline to analyze the expense of attempting to reduce one objective on the other, thus be able to better decide on the schedule to be implemented.

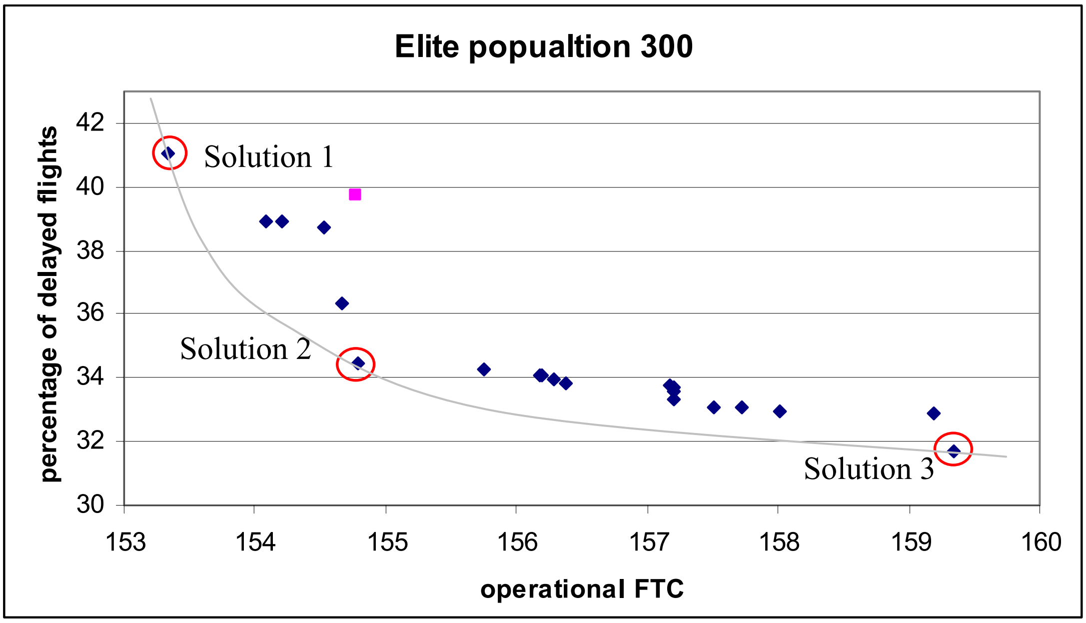



**Figure 6.3 Comparing solutions in elite population 300 with the original flight schedule**

The elite solutions in the final population, population 300 will be analyzed and is displayed in Figure 6.3. In particular, three solutions on the solution front are selected and highlighted in the figure. Solution 1 and Solution 3 correspond to two extreme results on the trade-off surface. By implementing Solution 1, the minimum average operational FTC of 153.34 can be achieved. Though this solution is able to achieve a low operational FTC, it is realized with a high percentage of delayed flights of 41.06%.

Conversely, implementation of Solution 3 will lead to a minimum average percentage delay of 31.69 at the expense of a high operational FTC of 159.34.

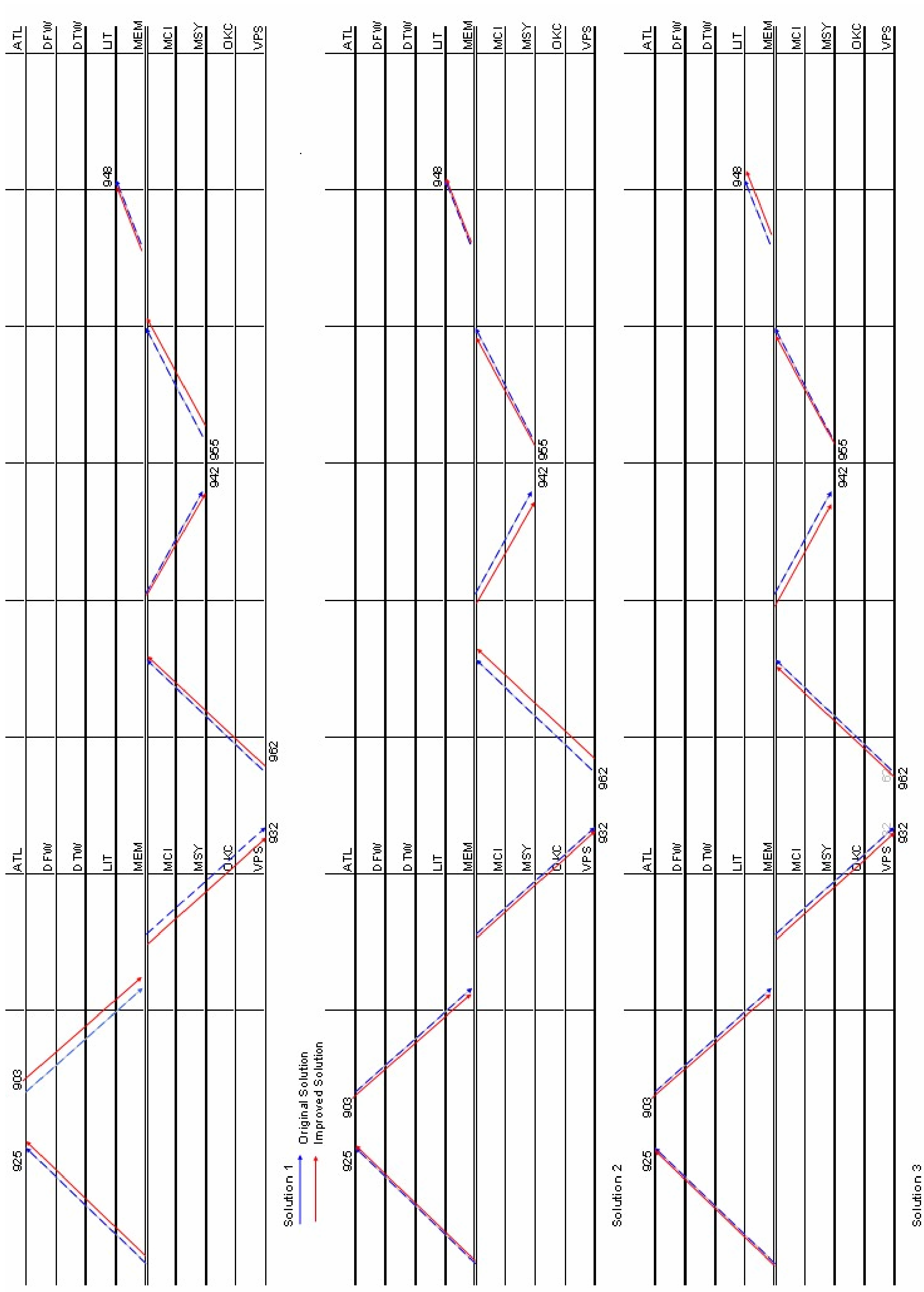



**Figure 6.4 Rotation 002 of test flight schedule**

### 6.1.2 Performance of percentage of flights delayed

To be able to analyze how an average minimum percentage of flights delayed is achieved by Solution 3, a particular rotation (rotation 002) is extracted and shown in Figure 6.4. The sequence of flights in this rotation is listed in Table 6-1. Under each set of solution, the "*depT*" is the new departure time as suggested by the solution. "*shift*" is the shift in the departure time (in minutes) for that flight, a positive value indicates that the flight is shifted to a time later than the original and vice versa. "$\Delta_{slack}$" is the change in slack between the previous flight and the current flight flown by this aircraft. A positive value indicates that the slack time between the flights is increased and vice versa. For example, in Solution 1, flight 925 is shifted later by two minutes and flight 903 is shifted later by four minutes; hence, the overall change in the slack time between flights 925 and 903 is an increase by two minutes.

**Table 6-1 Sequence of flight in rotation 002**

| | Original | Solution 1 | | | Solution 2 | | | Solution 3 | | |
|---|---|---|---|---|---|---|---|---|---|---|
| flight | *depT* | *depT* | *shift* | $\Delta_{slack}$ | *depT* | *shift* | $\Delta_{slack}$ | *depT* | *shift* | $\Delta_{slack}$ |
| 925 | 0839 | 0841 | 2 | | 0839 | 0 | | 0839 | 0 | |
| 903 | 1140 | 1144 | 4 | 2 | 1138 | -2 | -2 | 1138 | -2 | -2 |
| 932 | 1235 | 1231 | -4 | -8 | 1234 | -1 | 1 | 1233 | -2 | 0 |
| 962 | 1435 | 1436 | 1 | 5 | 1439 | 4 | 5 | 1433 | -2 | 0 |
| 942 | 1645 | 1641 | 0 | -1 | 1641 | -4 | -8 | 1640 | -5 | -3 |
| 955 | 1840 | 1844 | 4 | 4 | 1837 | -3 | 1 | 1838 | -2 | 3 |
| 948 | 2100 | 2100 | 0 | -4 | 2101 | 1 | 4 | 2104 | 4 | 6 |

To minimize the delay of flights, Solution 3 suggests shifting all the flights earlier, except flight 948 which it suggested to be shifted later in time. It may perhaps be due to the rush hour from 2000hrs to 2100hrs at the hub which often causes congestion. It has also been suggested that a delay at the start of the day could propagate along the network affecting the flights that will be flown by the same aircraft or same crew. Hence it is essential to augment the slack between these two flights.

Solution 3 clearly shows its attempt to minimize the delay by increasing the slack between flights 948 and 955 by 6 minutes. Solution 2 only increases the slack by 4 minutes and Solution 1 on the other hand, actually reduces the slack time. Moreover, the "$\Delta_{slack}$" column in Table 6-1 for Solution 3 shows a general trend such that the change in slack between flights increases as the departure time of the flights get closer to the rush hour. Solution 2 does have a similar trend although the increase is not as stable. This trend is clearly lacking in Solution 1. This reinforces the point that delay propagates along the network, causing more disruption to flight further down the network. To minimize the propagation of delay, more slack must be provided between all the flights in the rotation when congestion is expected to take place in the later part of the day. Thus, Solution 3 attempts to leave as much time allowance as possible between flights 955 and 948, resulting in flight 948 being shifted later and flight 955 earlier, all the other flights that precede flight 955 in the plane's rotation have also been shifted to an earlier time as a result. These can be observed from the flight network displayed in Figure 6.4. Solution 1 however, does not exhibit this trend.

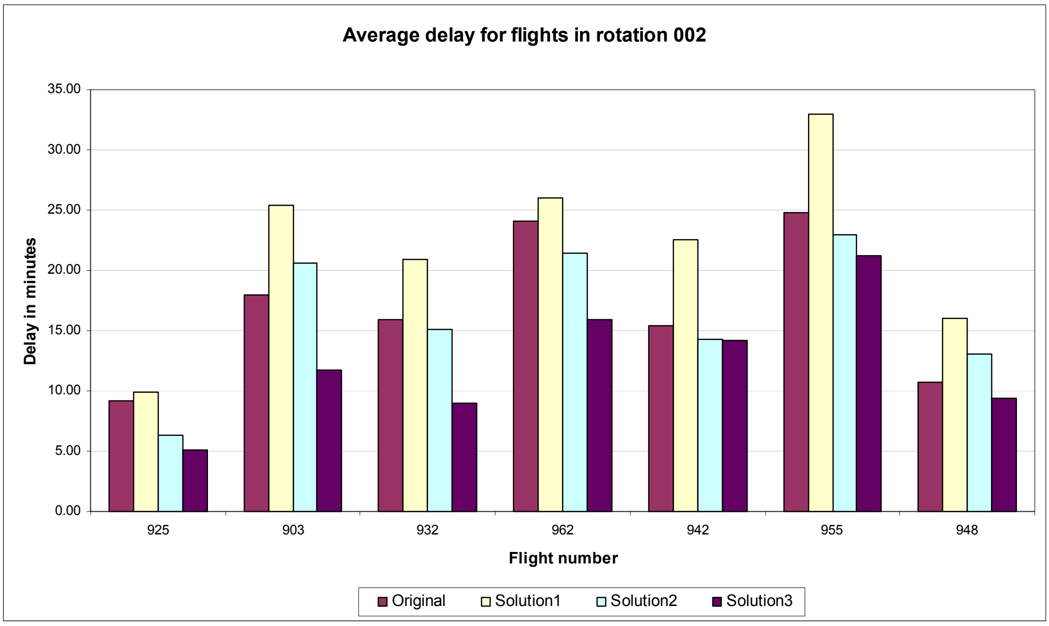



**Figure 6.5 Comparing the average delay of flights in rotation 002**

Figure 6.5 shows the distribution of the average delay for each of the flights in rotation 002. It is not difficult to notice that for every flight, the delay for Solution 3 is smaller than Solution 2, which is in turn, lesser than Solution 1. Clearly, Solution 3 outperforms Solution 1, Solution 2 and the original solution as well. It is also obvious that Solution 1 is not an ideal solution if minimum delay is to be achieved as the delay for these flights are greater as compared to the original schedule.

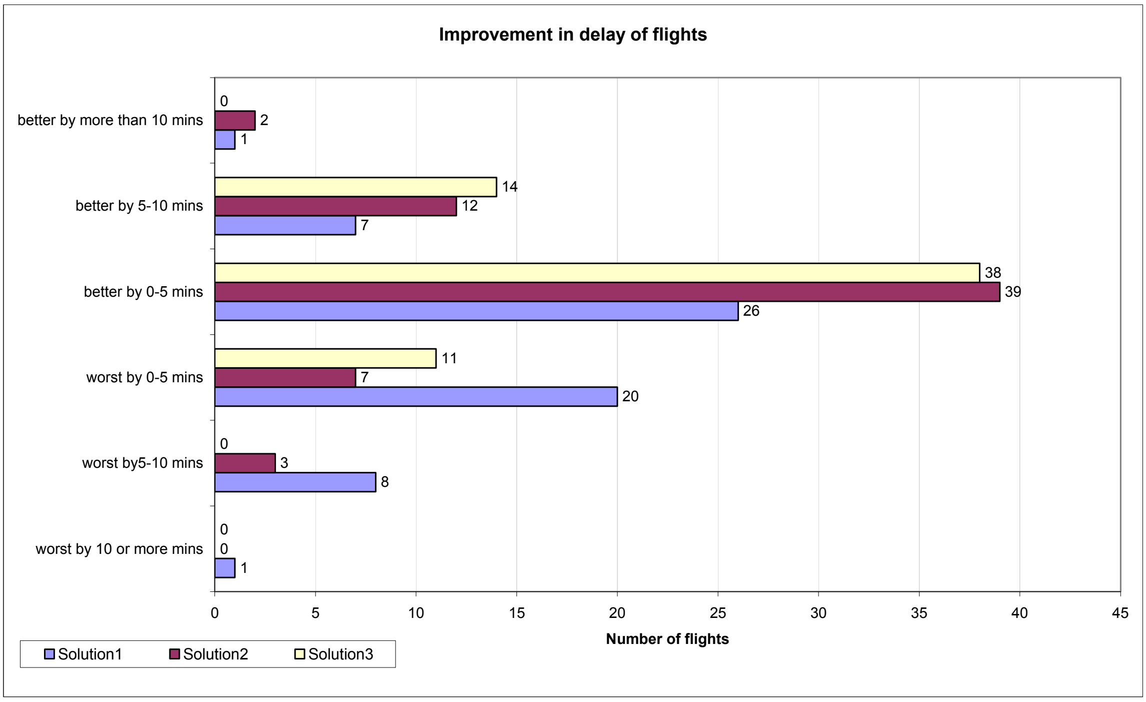



**Figure 6.6 Improvement in the delay of flights**

Figure 6.6 on the other hand, analyzes the amount of improvement in terms of delay for all the flights in the schedule (for each solution). This improvement is computed by comparing the new solutions with the delay in the original solution. Although Solution 3 does not have any flights that are better than the original by more than 10 minutes, a total of 52 flights out of 63 flights have lesser delay as compared to the original. This solution also does not have any flights that are worst by more than 5 minutes as compared to the original. Solution 2 has 53 flights with reduced delay, however, three of its flights have delay that are worst than the original by more than 5 minutes. From the plot of the elite population in Figure 6.3, the difference between the average percentage of flights delayed for Solution 2 and Solution 3 is only approximately 2%, thus it is not surprising that the performance of Solution 2 in terms of the improvement in delay of flights is just as good as that of Solution 3. However, the difference between the average percentage of flights delayed for Solution 1 as compare to Solution 2 and Solution 3 are 7% and 9% respectively, thus it is easy to understand that Solution 1 has many flights that have delays of longer duration as compared to the original solution.

Further analysis was carried out on the three selected solutions, by examining the tardiness of the flights. Tardiness can be defined as the quality of not adhering to the scheduled time, in other words, the lateness of the flights. A plot of the percentage of flights late for each of the five minute interval up to 100 minutes is illustrated in Figure 6.7. On the horizontal axis is the average delay of the flights in minutes, as the tardiness is of concern here, an early flight is equivalent to zero minutes late, rather than being assigned a negative value. The last column is for the flights that have been cancelled, these flight are also considered late. The vertical axis shows the percentage of flights that have been delayed for each of the solutions including the original flight schedule. Also shown on the same figure, is the cumulate distribution of the percentage of delay of the flights for each of five-minute interval.

Solutions 2 and 3 show improvements of more than 5% for the flights that arrive earlier than the scheduled time as shown in the first column. From the cumulative distribution plot, it is also clear that Solution 2 and Solution 3 has a higher proportion of flights with delay of less than 15 minutes, showing their superiority in achieving a minimum average delay.

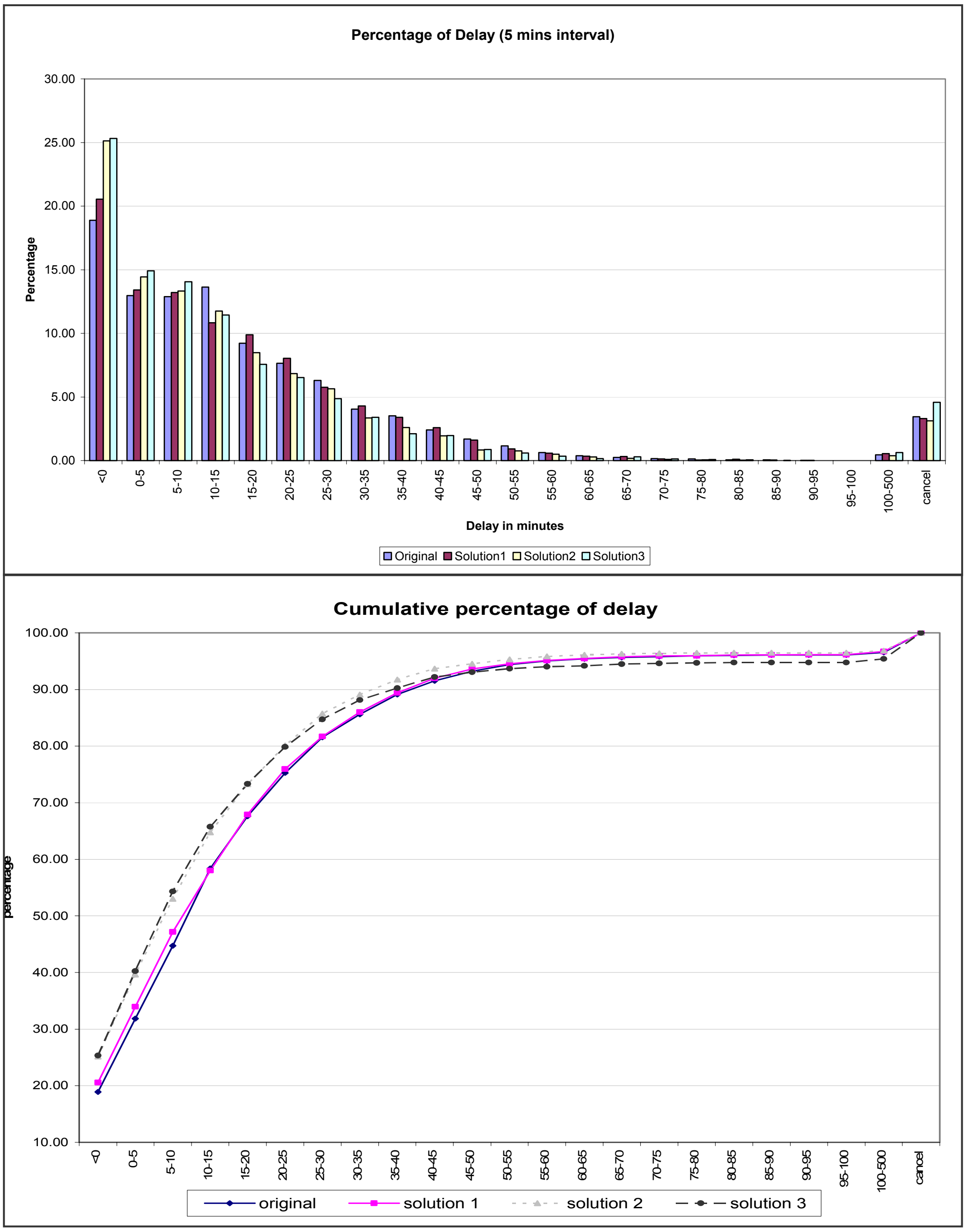



**Figure 6.7 Percentage of flights delayed against the delay in minutes (Top). Cumulative percentage of delay in minutes for different solutions (Bottom)**

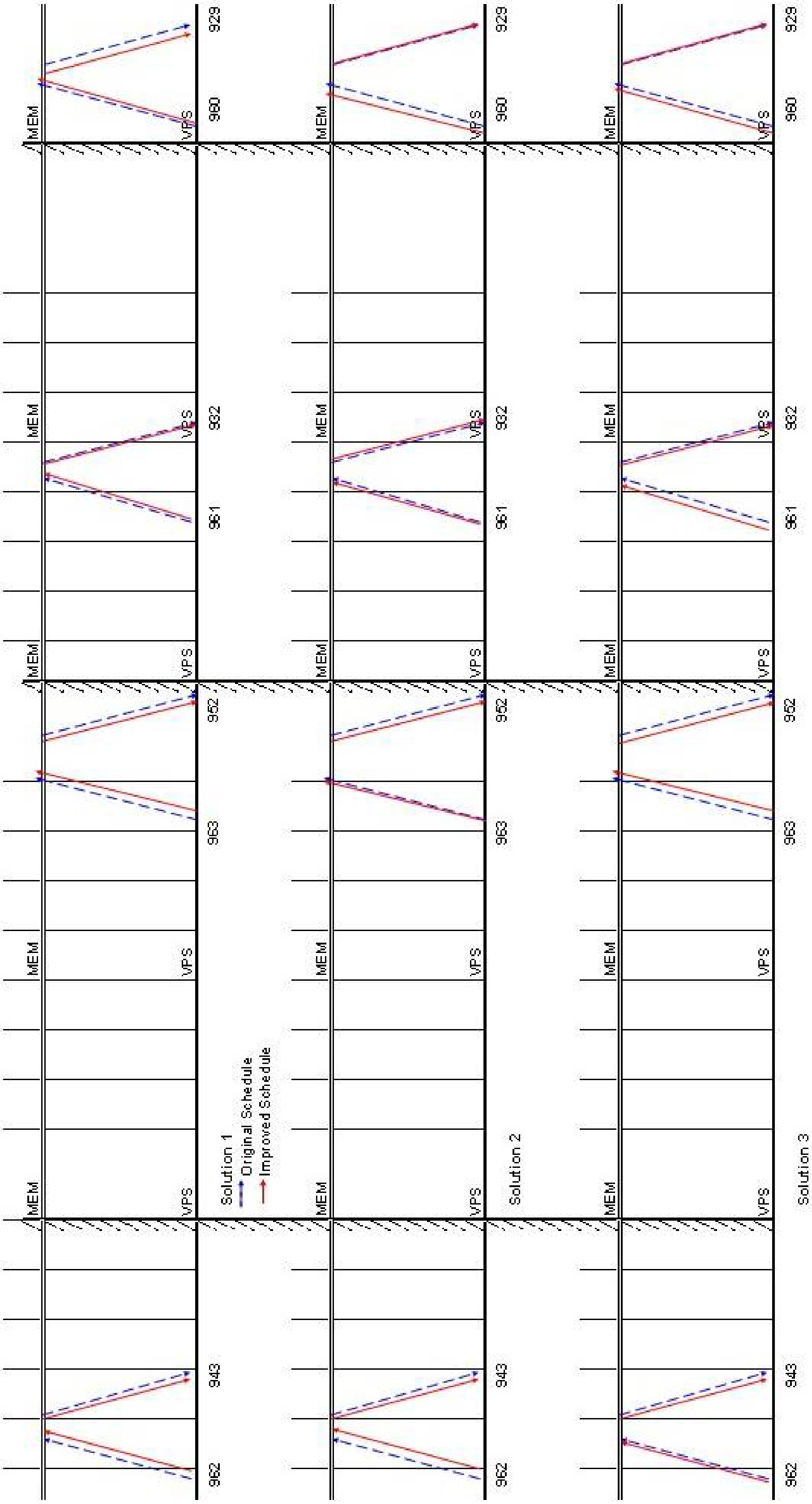



**Figure 6.8 Comparing the shift between the original schedule and the improved schedule for crew pairing 1907**

### 6.1.3 Performance of operational FTC

Figure 6.8 is an extraction of a four day long crew pairing, pairing 1907, which is used to analyze how operational FTC is minimized in Solution 1. The flights which are flown by the crew are listed in Table 6-2. Under each set of solution, the "*depT*" is the new departure time as suggested by the solution. "*shift*" is the shift in the departure time (in minutes) for that flight, a positive value indicates that the flight is shifted to a time later than the original and vice versa. The crew starts off with flight 962 and flight 943 on the first day, followed by 963 and 952 on the second day, 961 and 932 on the third and ends off with flight 960 and 929 on the fourth day.

**Table 6-2 The sequence of flights in crew pairing 1907**

| | Original | | Solution 1 | | Solution 2 | | Solution 3 | |
|---|---|---|---|---|---|---|---|---|
| flight | day of dep | *depT* | *depT* | *shift* | *depT* | *shift* | *depT* | *shift* |
| 962 | 1 | 1425 | 1426 | 1 | 1429 | 4 | 1423 | -2 |
| 943 | 1 | 1645 | 1641 | -4 | 1642 | -3 | 1642 | -3 |
| 963 | 2 | 1840 | 1842 | 2 | 1836 | -4 | 1842 | 2 |
| 952 | 2 | 2125 | 2121 | -4 | 2119 | -6 | 2118 | -7 |
| 961 | 3 | 1040 | 1040 | 0 | 1037 | -3 | 1036 | -4 |
| 932 | 3 | 1235 | 1231 | -4 | 1234 | -1 | 1233 | -2 |
| 960 | 4 | 0640 | 0645 | 5 | 0637 | -3 | 0638 | -2 |
| 929 | 4 | 0840 | 0836 | -4 | 0841 | 1 | 0840 | 0 |
| duration of TAFB | | | reduced by 5 | | reduced by 3 | | increased by 2 | |

As defined previously, FTC is the difference between the number of minutes paid and the number of minutes flown, as a percentage of the number of minutes flown. Since our solution procedure does not alter the block time of flight, the number of minutes flown remains constant for different solutions, leaving number of minutes paid the only variable across solutions. The number of minutes paid is a sum of all the pairing cost. Each pairing cost is in turn dependent on three components, the sum of all the duty cost, the time away from base (*TAFB*) and the number of duties (*numduties*); It takes on the value of the largest component such that

$$c_p = \max\left\{\sum_{d\in p} b_d, r_t \times TAFB_p, mgp \times numduties_p\right\}$$

$b_d$ is the cost of duty *d*. TAFB*p* is the time away from base of pairing *p* and $r_t$ is a fraction representing the rate of pay of TAFB. *mgp* is a minimum guarantee per duty in a pairing, and numduties$_p$ be the number of duties in pairing *p*. (The complete method of computing FTC has been discussed in a previous section, some of it is repeated here for the reader's convenience) Using the values suggested by Vance et al. (1997), $r_t = \frac{2}{7}$, and $mgp = 300$. For the original schedule,

$$c_p = \max\{1279, 1567, 1200\}$$

It is evident that the value of the second component, *TAFB* far exceeds the other two and is most likely to be selected as the minutes paid. To minimize the FTC Solution 1 suggest shifting flight 962 later by 1 minute and flight 929 earlier by 4 minutes, by doing so, the *TAFB* will be reduced by a total of five minutes, lowering the FTC. From the table, Solution 2 also reduces the *TAFB* but by a smaller amount of three minutes, whilst Solution 3 increases the *TAFB* by two minutes and hence resulting in a poorer FTC performance.

## 6.2 Test Data *B*

The same solution procedure was implemented on Test Data *B* for 300 generations, using the same set of procedure parameters. The plots of the elite populations are shown in Figure 6.9. In the same way, the elite population shifts leftwards and downwards after generations, verifying that the solution procedure is able to generated better solutions as compare to the original schedule.

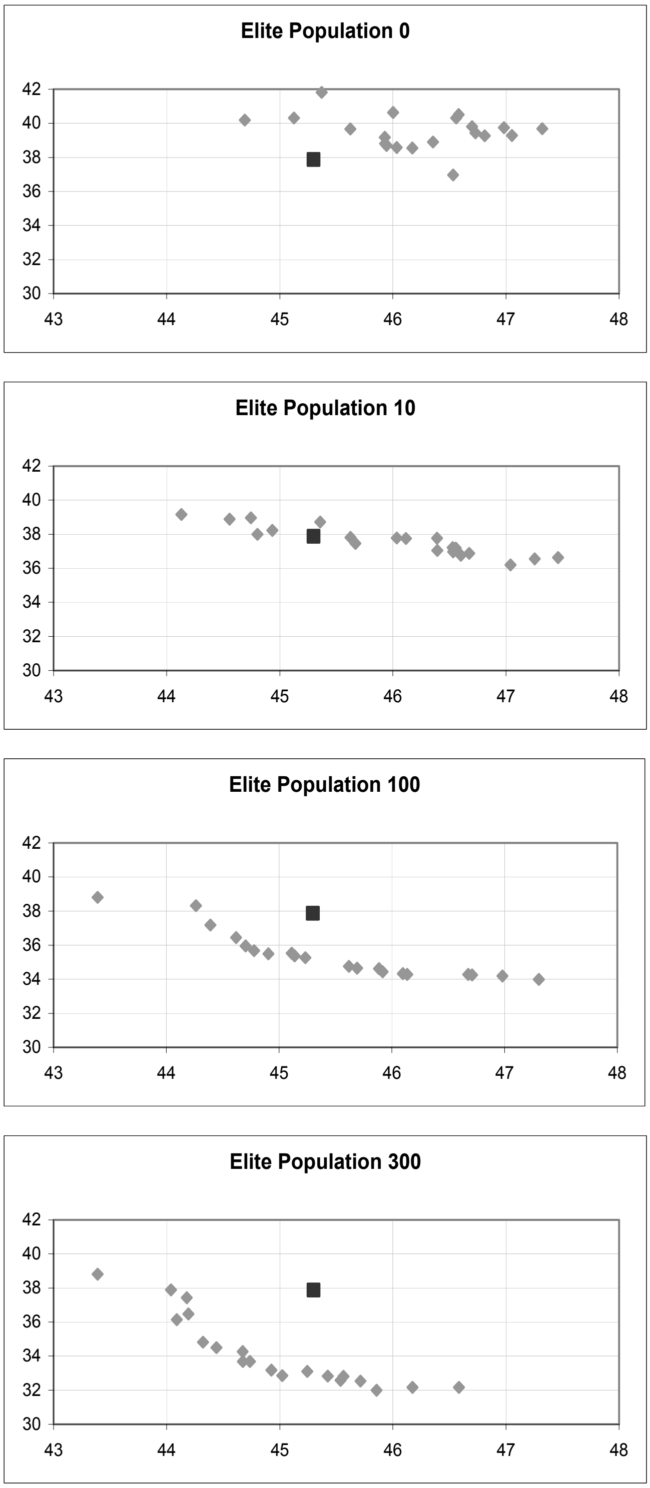



**Figure 6.9 Progression of elite population for Test Data *B***

For Test Data *B*, the elite population for the 300$^{th}$ generation are extracted and shown in Figure 6.10 together with the same elite population of Test Data *A*. In the same way, the solution procedure is able to improve on the given flight schedule of test data *B*. As observed for Test Data *A*, although none of the solutions in the elite population are inferior to the originally provided solution (i.e. the original solution does not dominate any of the generated solutions), only three out of the twenty elite solutions in Test Data *A* dominate the original flight schedule. For Test Data *B*, the elite population for generation 300 has generated more than 10 solutions that dominate the original flight schedule. The solution procedure has generated better results after 300 generations for Test Data *B*. It is observed that in both sets of solutions, most of the newly generated improved solutions have better performance in terms of percentage delay. However, comparing the performance of FTC, only a few solutions are better in FTC for Test Data *A*, while more than half of the solutions have lower FTC for Test Data *B*. Thus, most of the improvements for Test Data *B* were achieved due to a reduction in FTC. A probable reason of this result is because on average, there are more flight legs in each duty as compare to Test Data *A*, thus, there might be more room for improvement.

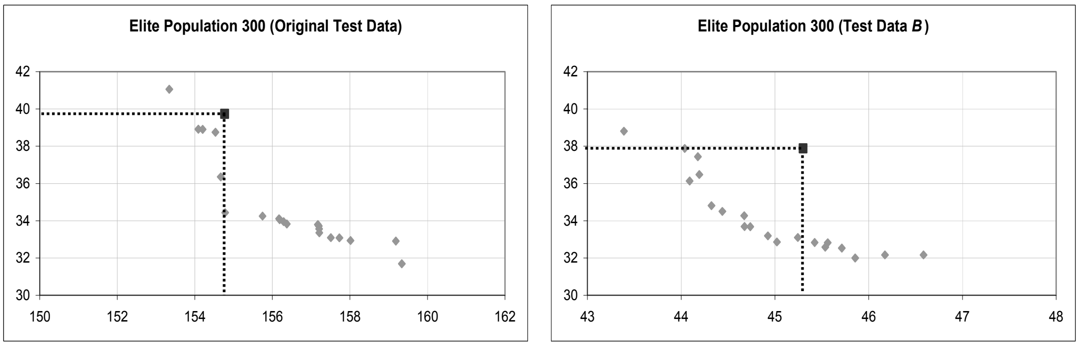



**Figure 6.10 300th Elite population for original test data and test data B**

## 6.2.1 Test Data *C*

The solution procedure was implemented on Test Data *C* for 300 generations, using the same set of procedure parameters that was used for Test Data *A*. The elite population for several generations is shown in Figure 6.11. From the progression of the elite population for the first 300 generations of Test Data *C*, it is clear that the solution procedure also works well when the FTC parameters are adjusted. Also, FTC seems to be rather insensitive to its parameters; the minimum FTC achieved is slightly more than 154, while the minimum FTC for the original test data is just below 154.

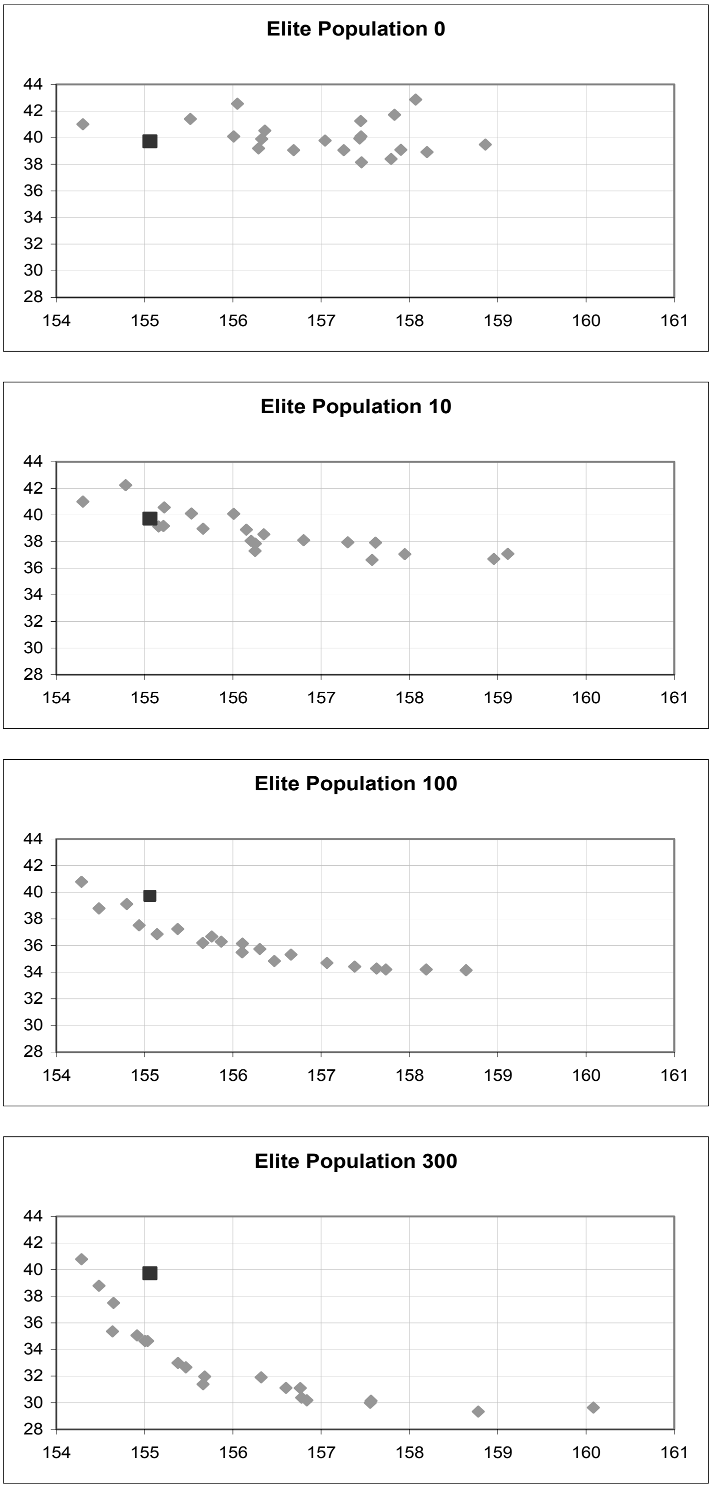



**Figure 6.11 Progression of the elite population for Test Data *C***

## 6.3 Summary

The solution procedure that was developed in this study to improve the robustness of a flight schedule was implemented on three set of flight data to minimize the operational FTC and percentage of flights delayed. For all three sets of test data, it was shown that the proposed solution procedure was capable obtaining flight schedules that are more robust than the given flight schedule in both of the objectives. By implementing the procedure, a set of non-dominated solution front can also be obtained. The average percentage of flights delayed for the original set of test data can be reduced by up to a maximum of approximately 8%, although the operational FTC can only be reduced by about 2%.

# CONCLUSION

In this thesis, the problem of airline flight schedules being highly sensitive to operational disturbances is addressed. A multi-objective model was formulated to improve the robustness of an existing flight schedule. By formulating the problem as a multi-objective problem, it allows airlines to optimize several different objectives simultaneously. An existing flight schedule and plan serves as an input to the procedure, our objective is to shift the departure times of each flight so as to achieve optimal performance. Thus, robustness of the flight schedule is accomplished by determining the optimal the departure times of each of the flight legs.

The solution procedure is built upon the idea of multi-objective genetic algorithms. By using genetic algorithms, a population of solutions can be obtained with a single run of the procedure as compared to using classical methods of solving multi-objective optimization problems, which results in only one solution each time.

At each generation of the procedure, a set of new flight schedules are generated; to evaluate these new schedules, SIMAIR, a simulation model that models airline operations is employed. For a solution to be robust, it has to perform relatively well in a number of different situations, implying that each schedule has to be evaluated several times, subjecting it to different scenarios each time. However, if this is done, it would be too computationally expensive. To deal the computation complexity of passing each solution through the simulation model several times, the concept of maintaining two populations is introduced. The first population is only evaluated once in each generation, and the best solutions are duplicated into the next generation to be evaluated repeatedly for different operational situations.

The solution is provided in the form of a set of non-dominated solutions, which forms the trade-off surface. The trade-off surface allows the airline to analyze the expense of attempting to reduce one objective on the other, thus be able to better decide on the schedule to be implemented. This solution procedure was implemented onto three sets of test data, minimizing the operational FTC and average percentage of flights delayed and it was shown that the procedure was able to generate solutions that that were better in both objectives.

This thesis has focused on developing a solution framework and procedure for improving the robustness of an existing flight schedule by shifting the departure times of each of the flights in the schedule. It was assumed that the aircraft rotation and the crew rosters were fixed. Future research into robust flight schedules can investigate on how improvements can be made to the schedule by allowing cancellation of flight legs, rerouting of aircraft and even rerouting of crew. It can be realized with an integrated airline schedule plan and an integrated recovery policy that considers the flight schedule, aircraft rotation and crew pairing simultaneously.

An important factor that we have left out in this study is whether the proposed new departure times will have as high a demand as the previous departure time. As noted in wells (1997), saleability of a flight is highly sensitive to differences in departure times. Thus, further research can be studied to enhance the model by considering the demand for each flight at specific times.

# Appendix A

**Simair.ini file**

```
;This is simair configuration file

[Parameters]
;SIMULATION_DAY=14
SIMULATION_DAY=7
NUM_OF_REPLICATIONS=1

;mode is a enmu type
;0 is stochastic case
;1 is deterministic case
;2 is semi case
MODE=0
;random number seed, if not specified it is 1 by default
RANDOM_SEED=216


TRACE_STATION=1
TRACE_EVENT=1
;start time of simulation
START_TIME=20030505
;performance data collection will only begin after WARM_UP_PERIOD of days simulation elapsed
WARM_UP_PERIOD=2

;Input Files Related
;set this to "0" if wishes to skip verifier checking, else set 1
VERIFIER_CHECK=1
;set this to "0" if wishes to revert back to old leg input format. Default is SSIM format
SSIM_FORMAT=0



[Input Files]
;STATION_FILE=../filesNW2/StationType.txt
STATION_FILE=../filesNW2/StationTypeCurfew.txt
FLIGHT_FILE=../filesNW2/legs.txt
;FLIGHT_FILE=../filesNW2/SSIMlegs.txt
PLANE_FILE=../filesNW2/planes.txt
ROTATION_FILE=../filesNW2/rotations.txt
SERVICE_STATE_FILE=../filesNW2/servicerate.txt
DISTRIBUTION_FILE=../filesNW2/distributions.txt
MAINTENANCE_FILE=../filesNW2/maintenance.txt
ALL_STATION_FILE=../filesNW2/allStations.txt
CREWBASE_FILE=../filesNW2/crewbase.txt
```

DUTY_FILE=../filesNW2/duty.txt
PAIRING_FILE=../filesNW2/pairings.txt
REGULARCREW_FILE=../filesNW2/regularCrew.txt
RULECONFIG_FILE=../filesNW2/ruleConfig.txt

[Output Files]
EVENT_LOG_DIR=../filesNW2/event/
STATION_LOG_DIR=../filesNW2/station/
LEGALITY_LOG_DIR=../filesNW2/legality/
PERFMEASURE_LOG_DIR=../filesNW2/performance/

---

**allStations.txt file**

| | | | |
|---|---|---|---|
| Alanta | USA | ATL | -0400 |
| Dallas | USA | DFW | -0500 |
| Detroit | USA | DTW | -0400 |
| LittleRock | USA | LIT | -0500 |
| Mci city | USA | MCI | -0500 |
| Memphis | USA | MEM | -0500 |
| Msy city | USA | MSY | -0500 |
| Oklahoma | USA | OKC | -0500 |
| Vps city | USA | VPS | -0500 |

---

**Crewbase.txt file**

| | | |
|---|---|---|
| 801 | 1 | DFW |
| 802 | 1 | LIT |
| 803 | 1 | MCI |
| 804 | 1 | MSY |
| 805 | 1 | OKC |
| 806 | 1 | VPS |
| 807 | 1 | DTW |
| 808 | 1 | ATL |
| 809 | 1 | MEM |

---

**duty.txt file**

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1101 | 3 | 0 | 907 | DFW | 1 | 924 | MEM | 1 | 908 | DFW | 1 | | | |
| 1102 | 3 | 0 | 938 | MEM | 1 | 909 | DFW | 1 | 946 | MEM | 1 | | | |
| 1201 | 2 | 0 | 916 | LIT | 1 | 935 | MEM | 1 | | | | | | |
| 1202 | 4 | 0 | 917 | LIT | 1 | 944 | MEM | 1 | 918 | LIT | 1 | 948 | MEM | 1 |
| 1301 | 4 | 0 | 919 | MCI | 1 | 927 | MEM | 1 | 920 | MCI | 1 | 933 | MEM | 1 |
| 1302 | 4 | 0 | 921 | MCI | 1 | 941 | MEM | 1 | 922 | MCI | 1 | 951 | MEM | 1 |
| 1401 | 2 | 0 | 953 | MSY | 1 | 934 | MEM | 1 | | | | | | |
| 1402 | 4 | 0 | 954 | MSY | 1 | 942 | MEM | 1 | 955 | MSY | 1 | 947 | MEM | 1 |
| 1501 | 4 | 0 | 956 | OKC | 1 | 928 | MEM | 1 | 957 | OKC | 1 | 931 | MEM | 1 |
| 1502 | 4 | 0 | 958 | OKC | 1 | 940 | MEM | 1 | 959 | OKC | 1 | 949 | MEM | 1 |
| 1601 | 4 | 0 | 960 | VPS | 1 | 929 | MEM | 1 | 961 | VPS | 1 | 932 | MEM | 1 |
| 1602 | 4 | 0 | 962 | VPS | 1 | 943 | MEM | 1 | 963 | VPS | 1 | 952 | MEM | 1 |
| 1701 | 2 | 0 | 913 | DTW | 1 | 945 | MEM | 1 | | | | | | |

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1801 | 2 | 0 | 902 | ATL | 1 | 911 | DTW | 1 | | | | | | |
| 1802 | 2 | 0 | 937 | MEM | 1 | 915 | DTW | 1 | | | | | |
| 1803 | 2 | 0 | 903 | ATL | 1 | 936 | MEM | 1 | | | | | |
| 1804 | 2 | 0 | 901 | ATL | 1 | 925 | MEM | 1 | | | | | |
| 1805 | 4 | 0 | 904 | ATL | 1 | 939 | MEM | 1 | 906 | ATL | 1 | 950 | MEM 1 |
| 1901 | 3 | 0 | 926 | MEM | 1 | 912 | DTW | 1 | 905 | ATL | 1 | | |
| 1902 | 2 | 0 | 923 | MEM | 1 | 910 | DTW | 1 | | | | | |
| 1903 | 2 | 0 | 930 | MEM | 1 | 914 | DTW | 1 | | | | | |
| 2101 | 3 | 0 | 907 | DFW | 2 | 924 | MEM | 2 | 908 | DFW | 2 | | |
| 2102 | 3 | 0 | 938 | MEM | 2 | 909 | DFW | 2 | 946 | MEM | 2 | | |
| 2201 | 2 | 0 | 916 | LIT | 2 | 935 | MEM | 2 | | | | | |
| 2202 | 4 | 0 | 917 | LIT | 2 | 944 | MEM | 2 | 918 | LIT | 2 | 948 | MEM 2 |
| 2301 | 4 | 0 | 919 | MCI | 2 | 927 | MEM | 2 | 920 | MCI | 2 | 933 | MEM 2 |
| 2302 | 4 | 0 | 921 | MCI | 2 | 941 | MEM | 2 | 922 | MCI | 2 | 951 | MEM 2 |
| 2401 | 2 | 0 | 953 | MSY | 2 | 934 | MEM | 2 | | | | | |
| 2402 | 4 | 0 | 954 | MSY | 2 | 942 | MEM | 2 | 955 | MSY | 2 | 947 | MEM 2 |
| 2501 | 4 | 0 | 956 | OKC | 2 | 928 | MEM | 2 | 957 | OKC | 2 | 931 | MEM 2 |
| 2502 | 4 | 0 | 958 | OKC | 2 | 940 | MEM | 2 | 959 | OKC | 2 | 949 | MEM 2 |
| 2601 | 4 | 0 | 960 | VPS | 2 | 929 | MEM | 2 | 961 | VPS | 2 | 932 | MEM 2 |
| 2602 | 4 | 0 | 962 | VPS | 2 | 943 | MEM | 2 | 963 | VPS | 2 | 952 | MEM 2 |
| 2701 | 2 | 0 | 913 | DTW | 2 | 945 | MEM | 2 | | | | | |
| 2801 | 2 | 0 | 902 | ATL | 2 | 911 | DTW | 2 | | | | | |
| 2802 | 2 | 0 | 937 | MEM | 2 | 915 | DTW | 2 | | | | | |
| 2803 | 2 | 0 | 903 | ATL | 2 | 936 | MEM | 2 | | | | | |
| 2804 | 2 | 0 | 901 | ATL | 2 | 925 | MEM | 2 | | | | | |
| 2805 | 4 | 0 | 904 | ATL | 2 | 939 | MEM | 2 | 906 | ATL | 2 | 950 | MEM 2 |
| 2901 | 3 | 0 | 926 | MEM | 2 | 912 | DTW | 2 | 905 | ATL | 2 | | |
| 2902 | 2 | 0 | 923 | MEM | 2 | 910 | DTW | 2 | | | | | |
| 2903 | 2 | 0 | 930 | MEM | 2 | 914 | DTW | 2 | | | | | |
| 3101 | 3 | 0 | 907 | DFW | 3 | 924 | MEM | 3 | 908 | DFW | 3 | | |
| 3102 | 3 | 0 | 938 | MEM | 3 | 909 | DFW | 3 | 946 | MEM | 3 | | |
| 3201 | 2 | 0 | 916 | LIT | 3 | 935 | MEM | 3 | | | | | |
| 3202 | 4 | 0 | 917 | LIT | 3 | 944 | MEM | 3 | 918 | LIT | 3 | 948 | MEM 3 |
| 3301 | 4 | 0 | 919 | MCI | 3 | 927 | MEM | 3 | 920 | MCI | 3 | 933 | MEM 3 |
| 3302 | 4 | 0 | 921 | MCI | 3 | 941 | MEM | 3 | 922 | MCI | 3 | 951 | MEM 3 |
| 3401 | 2 | 0 | 953 | MSY | 3 | 934 | MEM | 3 | | | | | |
| 3402 | 4 | 0 | 954 | MSY | 3 | 942 | MEM | 3 | 955 | MSY | 3 | 947 | MEM 3 |
| 3501 | 4 | 0 | 956 | OKC | 3 | 928 | MEM | 3 | 957 | OKC | 3 | 931 | MEM 3 |
| 3502 | 4 | 0 | 958 | OKC | 3 | 940 | MEM | 3 | 959 | OKC | 3 | 949 | MEM 3 |
| 3601 | 4 | 0 | 960 | VPS | 3 | 929 | MEM | 3 | 961 | VPS | 3 | 932 | MEM 3 |
| 3602 | 4 | 0 | 962 | VPS | 3 | 943 | MEM | 3 | 963 | VPS | 3 | 952 | MEM 3 |
| 3701 | 2 | 0 | 913 | DTW | 3 | 945 | MEM | 3 | | | | | |
| 3801 | 2 | 0 | 902 | ATL | 3 | 911 | DTW | 3 | | | | | |
| 3802 | 2 | 0 | 937 | MEM | 3 | 915 | DTW | 3 | | | | | |
| 3803 | 2 | 0 | 903 | ATL | 3 | 936 | MEM | 3 | | | | | |
| 3804 | 2 | 0 | 901 | ATL | 3 | 925 | MEM | 3 | | | | | |
| 3805 | 4 | 0 | 904 | ATL | 3 | 939 | MEM | 3 | 906 | ATL | 3 | 950 | MEM 3 |
| 3901 | 3 | 0 | 926 | MEM | 3 | 912 | DTW | 3 | 905 | ATL | 3 | | |
| 3902 | 2 | 0 | 923 | MEM | 3 | 910 | DTW | 3 | | | | | |
| 3903 | 2 | 0 | 930 | MEM | 3 | 914 | DTW | 3 | | | | | |

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 4101 | 3 | 0 | 907 | DFW | 4 | 924 | MEM | 4 | 908 | DFW | 4 | | | |
| 4102 | 3 | 0 | 938 | MEM | 4 | 909 | DFW | 4 | 946 | MEM | 4 | | | |
| 4201 | 2 | 0 | 916 | LIT | 4 | 935 | MEM | 4 | | | | | | |
| 4202 | 4 | 0 | 917 | LIT | 4 | 944 | MEM | 4 | 918 | LIT | | 948 | MEM | 4 |
| 4301 | 4 | 0 | 919 | MCI | 4 | 927 | MEM | 4 | 920 | MCI | | 933 | MEM | 4 |
| 4302 | 4 | 0 | 921 | MCI | 4 | 941 | MEM | 4 | 922 | MCI | | 951 | MEM | 4 |
| 4401 | 2 | 0 | 953 | MSY | 4 | 934 | MEM | 4 | | | | | | |
| 4402 | 4 | 0 | 954 | MSY | 4 | 942 | MEM | 4 | 955 | MSY | 4 | 947 | MEM | 4 |
| 4501 | 4 | 0 | 956 | OKC | 4 | 928 | MEM | 4 | 957 | OKC | 4 | 931 | MEM | 4 |
| 4502 | 4 | 0 | 958 | OKC | 4 | 940 | MEM | 4 | 959 | OKC | 4 | 949 | MEM | 4 |
| 4601 | 4 | 0 | 960 | VPS | 4 | 929 | MEM | 4 | 961 | VPS | 4 | 932 | MEM | 4 |
| 4602 | 4 | 0 | 962 | VPS | 4 | 943 | MEM | 4 | 963 | VPS | 4 | 952 | MEM | 4 |
| 4701 | 2 | 0 | 913 | DTW | 4 | 945 | MEM | 4 | | | | | | |
| 4801 | 2 | 0 | 902 | ATL | 4 | 911 | DTW | 4 | | | | | | |
| 4802 | 2 | 0 | 937 | MEM | 4 | 915 | DTW | 4 | | | | | | |
| 4803 | 2 | 0 | 903 | ATL | 4 | 936 | MEM | 4 | | | | | | |
| 4804 | 2 | 0 | 901 | ATL | 4 | 925 | MEM | 4 | | | | | | |
| 4805 | 4 | 0 | 904 | ATL | 4 | 939 | MEM | 4 | 906 | ATL | 4 | 950 | MEM | 4 |
| 4901 | 3 | 0 | 926 | MEM | 4 | 912 | DTW | 4 | 905 | ATL | 1 | 4 | | |
| 4902 | 2 | 0 | 923 | MEM | 4 | 910 | DTW | 4 | | | | | | |
| 4903 | 2 | 0 | 930 | MEM | 4 | 914 | DTW | 4 | | | | | | |
| 5101 | 3 | 0 | 907 | DFW | 5 | 924 | MEM | 5 | 908 | DFW | 5 | | | |
| 5102 | 3 | 0 | 938 | MEM | 5 | 909 | DFW | 5 | 946 | MEM | 5 | | | |
| 5201 | 2 | 0 | 916 | LIT | 5 | 935 | MEM | 5 | | | | | | |
| 5202 | 4 | 0 | 917 | LIT | 5 | 944 | MEM | 5 | 918 | LIT | 5 | 948 | MEM | 5 |
| 5301 | 4 | 0 | 919 | MCI | 5 | 927 | MEM | 5 | 920 | MCI | 5 | 933 | MEM | 5 |
| 5302 | 4 | 0 | 921 | MCI | 5 | 941 | MEM | 5 | 922 | MCI | 5 | 951 | MEM | 5 |
| 5401 | 2 | 0 | 953 | MSY | 5 | 934 | MEM | 5 | | | | | | |
| 5402 | 4 | 0 | 954 | MSY | 5 | 942 | MEM | 5 | 955 | MSY | 5 | 947 | MEM | 5 |
| 5501 | 4 | 0 | 956 | OKC | 5 | 928 | MEM | 5 | 957 | OKC | 5 | 931 | MEM | 5 |
| 5502 | 4 | 0 | 958 | OKC | 5 | 940 | MEM | 5 | 959 | OKC | 5 | 949 | MEM | 5 |
| 5601 | 4 | 0 | 960 | VPS | 5 | 929 | MEM | 5 | 961 | VPS | 5 | 932 | MEM | 5 |
| 5602 | 4 | 0 | 962 | VPS | 5 | 943 | MEM | 5 | 963 | VPS | 5 | 952 | MEM | 5 |
| 5701 | 2 | 0 | 913 | DTW | 5 | 945 | MEM | 5 | | | | | | |
| 5801 | 2 | 0 | 902 | ATL | 5 | 911 | DTW | 5 | | | | | | |
| 5802 | 2 | 0 | 937 | MEM | 5 | 915 | DTW | 5 | | | | | | |
| 5803 | 2 | 0 | 903 | ATL | 5 | 936 | MEM | 5 | | | | | | |
| 5804 | 2 | 0 | 901 | ATL | 5 | 925 | MEM | 5 | | | | | | |
| 5805 | 4 | 0 | 904 | ATL | 5 | 939 | MEM | 5 | 906 | ATL | 5 | 950 | MEM | 5 |
| 5901 | 3 | 0 | 926 | MEM | 5 | 912 | DTW | 5 | 905 | ATL | 5 | | | |
| 5902 | 2 | 0 | 923 | MEM | 5 | 910 | DTW | 5 | | | | | | |
| 5903 | 2 | 0 | 930 | MEM | 5 | 914 | DTW | 5 | | | | | | |
| 6101 | 3 | 0 | 907 | DFW | 6 | 924 | MEM | 6 | 908 | DFW | 6 | | | |
| 6102 | 3 | 0 | 938 | MEM | 6 | 909 | DFW | 6 | 946 | MEM | 6 | | | |
| 6201 | 2 | 0 | 916 | LIT | 6 | 935 | MEM | 6 | | | | | | |
| 6202 | 4 | 0 | 917 | LIT | 6 | 944 | MEM | 6 | 918 | LIT | 6 | 948 | MEM | 6 |
| 6301 | 4 | 0 | 919 | MCI | 6 | 927 | MEM | 6 | 920 | MCI | 6 | 933 | MEM | 6 |
| 6302 | 4 | 0 | 921 | MCI | 6 | 941 | MEM | 6 | 922 | MCI | 6 | 951 | MEM | 6 |
| 6401 | 2 | 0 | 953 | MSY | 6 | 934 | MEM | 6 | | | | | | |
| 6402 | 4 | 0 | 954 | MSY | 6 | 942 | MEM | 6 | 955 | MSY | 6 | 947 | MEM | 6 |

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 6501 | 4 | 0 | 956 | OKC | 6 | 928 | MEM | 6 | 957 | OKC | 6 | 931 | MEM | 6 |
| 6502 | 4 | 0 | 958 | OKC | 6 | 940 | MEM | 6 | 959 | OKC | 6 | 949 | MEM | 6 |
| 6601 | 4 | 0 | 960 | VPS | 6 | 929 | MEM | 6 | 961 | VPS | 6 | 932 | MEM | 6 |
| 6602 | 4 | 0 | 962 | VPS | 6 | 943 | MEM | 6 | 963 | VPS | 6 | 952 | MEM | 6 |
| 6701 | 2 | 0 | 913 | DTW | 6 | 945 | MEM | 6 | | | | | | |
| 6801 | 2 | 0 | 902 | ATL | 6 | 911 | DTW | 6 | | | | | | |
| 6802 | 2 | 0 | 937 | MEM | 6 | 915 | DTW | 6 | | | | | | |
| 6803 | 2 | 0 | 903 | ATL | 6 | 936 | MEM | 6 | | | | | | |
| 6804 | 2 | 0 | 901 | ATL | 6 | 925 | MEM | 6 | | | | | | |
| 6805 | 4 | 0 | 904 | ATL | 6 | 939 | MEM | 6 | 906 | ATL | 6 | 950 | MEM | 6 |
| 6901 | 3 | 0 | 926 | MEM | 6 | 912 | DTW | 6 | 905 | ATL | 6 | | | |
| 6902 | 2 | 0 | 923 | MEM | 6 | 910 | DTW | 6 | | | | | | |
| 6903 | 2 | 0 | 930 | MEM | 6 | 914 | DTW | 6 | | | | | | |
| 7101 | 3 | 0 | 907 | DFW | 7 | 924 | MEM | 7 | 908 | DFW | 7 | | | |
| 7102 | 3 | 0 | 938 | MEM | 7 | 909 | DFW | 7 | 946 | MEM | 7 | | | |
| 7201 | 2 | 0 | 916 | LIT | 7 | 935 | MEM | 7 | | | | | | |
| 7202 | 4 | 0 | 917 | LIT | 7 | 944 | MEM | 7 | 918 | LIT | 7 | 948 | MEM | 7 |
| 7301 | 4 | 0 | 919 | MCI | 7 | 927 | MEM | 7 | 920 | MCI | 7 | 933 | MEM | 7 |
| 7302 | 4 | 0 | 921 | MCI | 7 | 941 | MEM | 7 | 922 | MCI | 7 | 951 | MEM | 7 |
| 7401 | 2 | 0 | 953 | MSY | 7 | 934 | MEM | 7 | | | | | | |
| 7402 | 4 | 0 | 954 | MSY | 7 | 942 | MEM | 7 | 955 | MSY | 7 | 947 | MEM | 7 |
| 7501 | 4 | 0 | 956 | OKC | 7 | 928 | MEM | 7 | 957 | OKC | 7 | 931 | MEM | 7 |
| 7502 | 4 | 0 | 958 | OKC | 7 | 940 | MEM | 7 | 959 | OKC | 7 | 949 | MEM | 7 |
| 7601 | 4 | 0 | 960 | VPS | 7 | 929 | MEM | 7 | 961 | VPS | 7 | 932 | MEM | 7 |
| 7602 | 4 | 0 | 962 | VPS | 7 | 943 | MEM | 7 | 963 | VPS | 7 | 952 | MEM | 7 |
| 7701 | 2 | 0 | 913 | DTW | 7 | 945 | MEM | 7 | | | | | | |
| 7801 | 2 | 0 | 902 | ATL | 7 | 911 | DTW | 7 | | | | | | |
| 7802 | 2 | 0 | 937 | MEM | 7 | 915 | DTW | 7 | | | | | | |
| 7803 | 2 | 0 | 903 | ATL | 7 | 936 | MEM | 7 | | | | | | |
| 7804 | 2 | 0 | 901 | ATL | 7 | 925 | MEM | 7 | | | | | | |
| 7805 | 4 | 0 | 904 | ATL | 7 | 939 | MEM | 7 | 906 | ATL | 7 | 950 | MEM | 7 |
| 7901 | 3 | 0 | 926 | MEM | 7 | 912 | DTW | 7 | 905 | ATL | 7 | | | |
| 7902 | 2 | 0 | 923 | MEM | 7 | 910 | DTW | 7 | | | | | | |
| 7903 | 2 | 0 | 930 | MEM | 7 | 914 | DTW | 7 | | | | | | |

**legs.txt file**

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1234567 | 0700 | ATL | 0717 | 0 | MEM | 901 | P | 77 |
| 1234567 | 0834 | ATL | 1030 | 0 | DTW | 902 | P | 116 |
| 1234567 | 1140 | ATL | 1156 | 0 | MEM | 903 | P | 76 |
| 1234567 | 1530 | ATL | 1550 | 0 | MEM | 904 | P | 80 |
| 1234567 | 1600 | ATL | 1620 | 0 | MEM | 905 | P | 80 |
| 1234567 | 1930 | ATL | 1949 | 0 | MEM | 906 | P | 79 |
| 1234567 | 0625 | DFW | 0751 | 0 | MEM | 907 | P | 86 |
| 1234567 | 1045 | DFW | 1210 | 0 | MEM | 908 | P | 85 |
| 1234567 | 1840 | DFW | 2005 | 0 | MEM | 909 | P | 85 |
| 1234567 | 1030 | DTW | 1137 | 0 | MEM | 910 | P | 127 |
| 1234567 | 1118 | DTW | 1214 | 0 | MEM | 911 | P | 116 |
| 1234567 | 1225 | DTW | 1433 | 0 | ATL | 912 | P | 128 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1234567 | 1430 | DTW | 1531 | 0 | MEM | 913 | P | 121 |
| 1234567 | 1900 | DTW | 2003 | 0 | MEM | 914 | P | 123 |
| 1234567 | 2050 | DTW | 2251 | 0 | ATL | 915 | P | 121 |
| 1234567 | 0700 | LIT | 0748 | 0 | MEM | 916 | P | 48 |
| 1234567 | 1445 | LIT | 1532 | 0 | MEM | 917 | P | 47 |
| 1234567 | 1900 | LIT | 1946 | 0 | MEM | 918 | P | 46 |
| 1234567 | 0620 | MCI | 0738 | 0 | MEM | 919 | P | 78 |
| 1234567 | 1045 | MCI | 1203 | 0 | MEM | 920 | P | 78 |
| 1234567 | 1451 | MCI | 1610 | 0 | MEM | 921 | P | 79 |
| 1234567 | 1850 | MCI | 2010 | 0 | MEM | 922 | P | 80 |
| 1234567 | 0620 | MEM | 0918 | 0 | DTW | 923 | P | 118 |
| 1234567 | 0825 | MEM | 0959 | 0 | DFW | 924 | P | 94 |
| 1234567 | 0835 | MEM | 1100 | 0 | ATL | 925 | P | 85 |
| 1234567 | 0839 | MEM | 1135 | 0 | DTW | 926 | P | 116 |
| 1234567 | 0840 | MEM | 1004 | 0 | MCI | 927 | P | 84 |
| 1234567 | 0840 | MEM | 1009 | 0 | OKC | 928 | P | 89 |
| 1234567 | 0840 | MEM | 0959 | 0 | VPS | 929 | P | 79 |
| 1234567 | 0945 | MEM | 1240 | 0 | DTW | 930 | P | 115 |
| 1234567 | 1245 | MEM | 1414 | 0 | OKC | 931 | P | 89 |
| 1234567 | 1235 | MEM | 1353 | 0 | VPS | 932 | P | 78 |
| 1234567 | 1245 | MEM | 1410 | 0 | MCI | 933 | P | 85 |
| 1234567 | 1245 | MEM | 1400 | 0 | MSY | 934 | P | 75 |
| 1234567 | 1250 | MEM | 1340 | 0 | LIT | 935 | P | 50 |
| 1234567 | 1255 | MEM | 1518 | 0 | ATL | 936 | P | 83 |
| 1234567 | 1300 | MEM | 1553 | 0 | DTW | 937 | P | 113 |
| 1234567 | 1630 | MEM | 1805 | 0 | DFW | 938 | P | 95 |
| 1234567 | 1635 | MEM | 1846 | 0 | ATL | 939 | P | 71 |
| 1234567 | 1635 | MEM | 1804 | 0 | OKC | 940 | P | 89 |
| 1234567 | 1645 | MEM | 1810 | 0 | MCI | 941 | P | 85 |
| 1234567 | 1645 | MEM | 1800 | 0 | MSY | 942 | P | 75 |
| 1234567 | 1645 | MEM | 1803 | 0 | VPS | 943 | P | 78 |
| 1234567 | 1700 | MEM | 1749 | 0 | LIT | 944 | P | 49 |
| 1234567 | 1720 | MEM | 2020 | 0 | DTW | 945 | P | 120 |
| 1234567 | 2045 | MEM | 2220 | 0 | DFW | 946 | P | 95 |
| 1234567 | 2050 | MEM | 2202 | 0 | MSY | 947 | P | 72 |
| 1234567 | 2100 | MEM | 2147 | 0 | LIT | 948 | P | 47 |
| 1234567 | 2100 | MEM | 2230 | 0 | OKC | 949 | P | 90 |
| 1234567 | 2105 | MEM | 2329 | 0 | ATL | 950 | P | 84 |
| 1234567 | 2105 | MEM | 2230 | 0 | MCI | 951 | P | 85 |
| 1234567 | 2125 | MEM | 2242 | 0 | VPS | 952 | P | 77 |
| 1234567 | 0625 | MSY | 0740 | 0 | MEM | 953 | P | 75 |
| 1234567 | 1440 | MSY | 1559 | 0 | MEM | 954 | P | 79 |
| 1234567 | 1840 | MSY | 1959 | 0 | MEM | 955 | P | 79 |
| 1234567 | 0625 | OKC | 0745 | 0 | MEM | 956 | P | 80 |
| 1234567 | 1050 | OKC | 1210 | 0 | MEM | 957 | P | 80 |
| 1234567 | 1444 | OKC | 1604 | 0 | MEM | 958 | P | 80 |
| 1234567 | 1845 | OKC | 2004 | 0 | MEM | 959 | P | 79 |
| 1234567 | 0640 | VPS | 0805 | 0 | MEM | 960 | P | 85 |
| 1234567 | 1040 | VPS | 1203 | 0 | MEM | 961 | P | 83 |
| 1234567 | 1435 | VPS | 1557 | 0 | MEM | 962 | P | 82 |

1234567 1840 VPS 2001 0 MEM 963 P 81

---

**maintenance.txt file**
AA 70 100 50000 95000 110 200 180 MEM DTW DFW

---

**Pairing.txt file**

| | | | | |
|---|---|---|---|---|
| 8101 | 1 | 0 | 1102 | |
| 8102 | 2 | 0 | 1101 | 2102 |
| 8103 | 2 | 0 | 2101 | 3102 |
| 8104 | 2 | 0 | 3101 | 4102 |
| 8105 | 2 | 0 | 4101 | 5102 |
| 8106 | 2 | 0 | 5101 | 6102 |
| 8107 | 2 | 0 | 6101 | 7102 |
| 8108 | 1 | 0 | 7101 | |
| 8201 | 1 | 0 | 1202 | |
| 8202 | 2 | 0 | 1201 | 2202 |
| 8203 | 2 | 0 | 2201 | 3202 |
| 8204 | 2 | 0 | 3201 | 4202 |
| 8205 | 2 | 0 | 4201 | 5202 |
| 8206 | 2 | 0 | 5201 | 6202 |
| 8207 | 2 | 0 | 6201 | 7202 |
| 8208 | 1 | 0 | 7201 | |
| 8301 | 1 | 0 | 1302 | |
| 8302 | 2 | 0 | 1301 | 2302 |
| 8303 | 2 | 0 | 2301 | 3302 |
| 8304 | 2 | 0 | 3301 | 4302 |
| 8305 | 2 | 0 | 4301 | 5302 |
| 8306 | 2 | 0 | 5301 | 6302 |
| 8307 | 2 | 0 | 6301 | 7302 |
| 8308 | 1 | 0 | 7301 | |
| 8401 | 1 | 0 | 1402 | |
| 8402 | 2 | 0 | 1401 | 2402 |
| 8403 | 2 | 0 | 2401 | 3402 |
| 8404 | 2 | 0 | 3401 | 4402 |
| 8405 | 2 | 0 | 4401 | 5402 |
| 8406 | 2 | 0 | 5401 | 6402 |
| 8407 | 2 | 0 | 6401 | 7402 |
| 8408 | 1 | 0 | 7401 | |
| 8501 | 1 | 0 | 1502 | |
| 8502 | 2 | 0 | 1501 | 2502 |
| 8503 | 2 | 0 | 2501 | 3502 |
| 8504 | 2 | 0 | 3501 | 4502 |
| 8505 | 2 | 0 | 4501 | 5502 |
| 8506 | 2 | 0 | 5501 | 6502 |
| 8507 | 2 | 0 | 6501 | 7502 |
| 8508 | 1 | 0 | 7501 | |
| 8601 | 1 | 0 | 1602 | |
| 8602 | 2 | 0 | 1601 | 2602 |

| | | | | | |
|---|---|---|---|---|---|
| 8603 | 2 | 0 | 2601 | 3602 | |
| 8604 | 2 | 0 | 3601 | 4602 | |
| 8605 | 2 | 0 | 4601 | 5602 | |
| 8606 | 2 | 0 | 5601 | 6602 | |
| 8607 | 2 | 0 | 6601 | 7602 | |
| 8608 | 1 | 0 | 7601 | | |
| 8701 | 3 | 0 | 1701 | 2701 | 3701 |
| 8702 | 4 | 0 | 4701 | 5701 | 6701 |
| 8801 | 1 | 0 | 1803 | | |
| 8802 | 2 | 0 | 1802 | 2803 | |
| 8803 | 3 | 0 | 1801 | 2802 | 3803 |
| 8804 | 3 | 0 | 2801 | 3802 | 4803 |
| 8805 | 3 | 0 | 3801 | 4802 | 5803 |
| 8806 | 3 | 0 | 4801 | 5802 | 6803 |
| 8807 | 3 | 0 | 5801 | 6802 | 7803 |
| 8808 | 2 | 0 | 6801 | 7802 | |
| 8809 | 1 | 0 | 7801 | | |
| 8810 | 1 | 0 | 1805 | | |
| 8811 | 2 | 0 | 1804 | 2805 | |
| 8812 | 2 | 0 | 2804 | 3805 | |
| 8813 | 2 | 0 | 3804 | 4805 | |
| 8814 | 2 | 0 | 4804 | 5805 | |
| 8815 | 2 | 0 | 5804 | 6805 | |
| 8816 | 2 | 0 | 6804 | 7805 | |
| 8817 | 1 | 0 | 7804 | | |
| 8901 | 1 | 0 | 1903 | | |
| 8902 | 2 | 0 | 1902 | 2903 | |
| 8903 | 3 | 0 | 1901 | 2902 | 3903 |
| 8904 | 3 | 0 | 2901 | 3902 | 4903 |
| 8905 | 3 | 0 | 3901 | 4902 | 5903 |
| 8906 | 3 | 0 | 4901 | 5902 | 6903 |
| 8907 | 3 | 0 | 5901 | 6902 | 7903 |
| 8908 | 2 | 0 | 6901 | 7902 | |
| 8909 | 1 | 0 | 7901 | | |

**planes.txt file**

| | | | | | |
|---|---|---|---|---|---|
| NA001 | P | 11 | 743 | 200 | AA 1 |
| NA002 | P | 11 | 743 | 200 | AA 2 |
| NA003 | P | 11 | 743 | 200 | AA 1 |
| NA004 | P | 11 | 743 | 200 | AA 2 |
| NA005 | P | 11 | 743 | 200 | AA 1 |
| NA006 | P | 11 | 743 | 200 | AA 2 |
| NA007 | P | 11 | 743 | 200 | AA 1 |
| NA008 | P | 11 | 743 | 200 | AA 2 |
| NA009 | P | 11 | 743 | 200 | AA 1 |

**regularCrew.txt file**

9101 11 A DFW 8101 8104 8107

| | | | | | | |
|---|---|---|---|---|---|---|
| 9102 | 11 | A | DFW | 8102 | 8105 | 8108 |
| 9103 | 11 | A | DFW | 8103 | 8106 | |
| 9201 | 11 | A | LIT | 8201 | 8204 | 8207 |
| 9202 | 11 | A | LIT | 8202 | 8205 | 8208 |
| 9203 | 11 | A | LIT | 8203 | 8206 | |
| 9301 | 11 | A | MCI | 8301 | 8304 | 8307 |
| 9302 | 11 | A | MCI | 8302 | 8305 | 8308 |
| 9303 | 11 | A | MCI | 8303 | 8306 | |
| 9401 | 11 | A | MSY | 8401 | 8404 | 8407 |
| 9402 | 11 | A | MSY | 8402 | 8405 | 8408 |
| 9403 | 11 | A | MSY | 8403 | 8406 | |
| 9501 | 11 | A | OKC | 8501 | 8504 | 8507 |
| 9502 | 11 | A | OKC | 8502 | 8505 | 8508 |
| 9503 | 11 | A | OKC | 8503 | 8506 | |
| 9601 | 11 | A | VPS | 8601 | 8604 | 8607 |
| 9602 | 11 | A | VPS | 8602 | 8605 | 8608 |
| 9603 | 11 | A | VPS | 8603 | 8506 | |
| 9701 | 11 | A | DTW | 8701 | | |
| 9702 | 11 | A | DTW | 8702 | | |
| 9801 | 11 | A | ATL | 8801 | 8808 | |
| 9901 | 11 | A | MEM | 8802 | 8809 | |
| 9802 | 11 | A | ATL | 8803 | | |
| 9803 | 11 | A | ATL | 8804 | | |
| 9804 | 11 | A | ATL | 8805 | | |
| 9805 | 11 | A | ATL | 8806 | | |
| 9806 | 11 | A | ATL | 8807 | | |
| 9807 | 11 | A | ATL | 8810 | 8813 | 8816 |
| 9808 | 11 | A | ATL | 8811 | 8814 | 8817 |
| 9809 | 11 | A | ATL | 8812 | 8815 | |
| 9902 | 11 | A | MEM | 8901 | 8908 | |
| 9903 | 11 | A | MEM | 8902 | 8909 | |
| 9904 | 11 | A | MEM | 8903 | | |
| 9905 | 11 | A | MEM | 8904 | | |
| 9906 | 11 | A | MEM | 8905 | | |
| 9907 | 11 | A | MEM | 8906 | | |
| 9908 | 11 | A | MEM | 8907 | | |
| 10101 | 11 | B | DFW | 8101 | 8104 | 8107 |
| 10102 | 11 | B | DFW | 8102 | 8105 | 8108 |
| 10103 | 11 | B | DFW | 8103 | 8106 | |
| 10201 | 11 | B | LIT | 8201 | 8204 | 8207 |
| 10202 | 11 | B | LIT | 8202 | 8205 | 8208 |
| 10203 | 11 | B | LIT | 8203 | 8206 | |
| 10301 | 11 | B | MCI | 8301 | 8304 | 8307 |
| 10302 | 11 | B | MCI | 8302 | 8305 | 8308 |
| 10303 | 11 | B | MCI | 8303 | 8306 | |
| 10401 | 11 | B | MSY | 8401 | 8404 | 8407 |
| 10402 | 11 | B | MSY | 8402 | 8405 | 8408 |
| 10403 | 11 | B | MSY | 8403 | 8406 | |
| 10501 | 11 | B | OKC | 8501 | 8504 | 8507 |
| 10502 | 11 | B | OKC | 8502 | 8505 | 8508 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 10503 | 11 | B | OKC | 8503 | 8506 | |
| 10601 | 11 | B | VPS | 8601 | 8604 | 8607 |
| 10602 | 11 | B | VPS | 8602 | 8605 | 8608 |
| 10603 | 11 | B | VPS | 8603 | 8506 | |
| 10701 | 11 | B | DTW | 8701 | | |
| 10702 | 11 | B | DTW | 8702 | | |
| 10801 | 11 | B | ATL | 8801 | 8808 | |
| 10901 | 11 | B | MEM | 8802 | 8809 | |
| 10802 | 11 | B | ATL | 8803 | | |
| 10803 | 11 | B | ATL | 8804 | | |
| 10804 | 11 | B | ATL | 8805 | | |
| 10805 | 11 | B | ATL | 8806 | | |
| 10806 | 11 | B | ATL | 8807 | | |
| 10807 | 11 | B | ATL | 8810 | 8813 | 8816 |
| 10808 | 11 | B | ATL | 8811 | 8814 | 8817 |
| 10809 | 11 | B | ATL | 8812 | 8815 | |
| 10902 | 11 | B | MEM | 8901 | 8908 | |
| 10903 | 11 | B | MEM | 8902 | 8909 | |
| 10904 | 11 | B | MEM | 8903 | | |
| 10905 | 11 | B | MEM | 8904 | | |
| 10906 | 11 | B | MEM | 8905 | | |
| 10907 | 11 | B | MEM | 8906 | | |
| 10908 | 11 | B | MEM | 8907 | | |

---

**rotation.txt file**

NA001 953 MSY 1 924 MEM 1 908 DFW 1 935 MEM 1 917 LIT 1 939
MEM 1 906 ATL 1 949 MEM 1 956 OKC 2 925 MEM 2 903 ATL 2 932
MEM 2 962 VPS 2 942 MEM 2 955 MSY 2 948 MEM 2 (AA) 916 LIT 3
926 MEM 3 912 DTW 3 904 ATL 3 940 MEM 3 959 OKC 3 951 MEM 3
919 MCI 4 928 MEM 4 957 OKC 4 937 MEM 4 914 DTW 4 946 MEM 4
907 DFW 4 (AA) 929 MEM 5 961 VPS 5 933 MEM 5 921 MCI 5 944
MEM 5 918 LIT 5 950 MEM 6 902 ATL 6 911 DTW 6 936 MEM 6 905 ATL
6 945 MEM 6 915 DTW 6 901 ATL 7 927 MEM 7 920 MCI 7 934 MEM 7
954 MSY 7 943 MEM 7 963 VPS 7
NA002 956 OKC 1 925 MEM 1 903 ATL 1 932 MEM 1 962 VPS 1 942
MEM 1 955 MSY 1 948 MEM 1 (AA) 916 LIT 2 926 MEM 2 912 DTW 2
904 ATL 2 940 MEM 2 959 OKC 2 951 MEM 2 919 MCI 3 928 MEM 3
957 OKC 3 937 MEM 3 914 DTW 3 946 MEM 3 907 DFW 3 (AA) 929
MEM 4 961 VPS 4 933 MEM 4 921 MCI 4 944 MEM 4 918 LIT 4 950
MEM 5 902 ATL 5 911 DTW 5 936 MEM 5 905 ATL 5 945 MEM 5 915
DTW 5 (AA) 901 ATL 6 927 MEM 6 920 MCI 6 934 MEM 6 954 MSY 6
943 MEM 6 963 VPS 6 923 MEM 7 910 DTW 7 931 MEM 7 958 OKC 7
941 MEM 7 922 MCI 7 952 MEM 7 (AA)
NA003 916 LIT 1 926 MEM 1 912 DTW 1 904 ATL 1 940 MEM 1 959
OKC 1 951 MEM 1 919 MCI 2 928 MEM 2 957 OKC 2 937 MEM 2 914
DTW 2 946 MEM 2 907 DFW 2 (AA) 929 MEM 3 961 VPS 3 933 MEM 3
921 MCI 3 944 MEM 3 918 LIT 3 950 MEM 4 902 ATL 4 911 DTW 4
936 MEM 4 905 ATL 4 945 MEM 4 915 DTW 4 (AA) 901 ATL 5 927

MEM 5 920 MCI 5 934 MEM 5 954 MSY 5 943 MEM 5 963 VPS 5 923 MEM 6 910 DTW 6 931 MEM 6 958 OKC 6 941 MEM 6 922 MCI 6 952 MEM 6 (AA) 960 VPS 7 930 MEM 7 913 DTW 7 938 MEM 7 909 DFW 7 947 MEM 7

NA004 919 MCI 1 928 MEM 1 957 OKC 1 937 MEM 1 914 DTW 1 946 MEM 1 907 DFW 1 (AA) 929 MEM 2 961 VPS 2 933 MEM 2 921 MCI 2 944 MEM 2 918 LIT 2 950 MEM 3 902 ATL 3 911 DTW 3 936 MEM 3 905 ATL 3 945 MEM 3 915 DTW 3 (AA) 901 ATL 4 927 MEM 4 920 MCI 4 934 MEM 4 954 MSY 4 943 MEM 4 963 VPS 4 923 MEM 5 910 DTW 5 931 MEM 5 958 OKC 5 941 MEM 5 922 MCI 5 952 MEM 5 (AA) 960 VPS 6 930 MEM 6 913 DTW 6 938 MEM 6 909 DFW 6 947 MEM 6 953 MSY 7 924 MEM 7 908 DFW 7 935 MEM 7 917 LIT 7 939 MEM 7 906 ATL 7 949 MEM 7 (AA)

NA005 929 MEM 1 961 VPS 1 933 MEM 1 921 MCI 1 944 MEM 1 918 LIT 1 950 MEM 2 902 ATL 2 911 DTW 2 936 MEM 2 905 ATL 2 945 MEM 2 915 DTW 2 (AA) 901 ATL 3 927 MEM 3 920 MCI 3 934 MEM 3 954 MSY 3 943 MEM 3 963 VPS 3 923 MEM 4 910 DTW 4 931 MEM 4 958 OKC 4 941 MEM 4 922 MCI 4 952 MEM 4 (AA) 960 VPS 5 930 MEM 5 913 DTW 5 938 MEM 5 909 DFW 5 947 MEM 5 953 MSY 6 924 MEM 6 908 DFW 6 935 MEM 6 917 LIT 6 939 MEM 6 906 ATL 6 949 MEM 6 (AA) 956 OKC 7 925 MEM 7 903 ATL 7 932 MEM 7 962 VPS 7 942 MEM 7 955 MSY 7 948 MEM 7

NA006 950 MEM 1 902 ATL 1 911 DTW 1 936 MEM 1 905 ATL 1 945 MEM 1 915 DTW 1 (AA) 901 ATL 2 927 MEM 2 920 MCI 2 934 MEM 2 954 MSY 2 943 MEM 2 963 VPS 2 923 MEM 3 910 DTW 3 931 MEM 3 958 OKC 3 941 MEM 3 922 MCI 3 952 MEM 3 (AA) 960 VPS 4 930 MEM 4 913 DTW 4 938 MEM 4 909 DFW 4 947 MEM 4 953 MSY 5 924 MEM 5 908 DFW 5 935 MEM 5 917 LIT 5 939 MEM 5 906 ATL 5 949 MEM 5 (AA) 956 OKC 6 925 MEM 6 903 ATL 6 932 MEM 6 962 VPS 6 942 MEM 6 955 MSY 6 948 MEM 6 916 LIT 7 926 MEM 7 912 DTW 7 904 ATL 7 940 MEM 7 959 OKC 7 951 MEM 7

NA007 901 ATL 1 927 MEM 1 920 MCI 1 934 MEM 1 954 MSY 1 943 MEM 1 963 VPS 1 923 MEM 2 910 DTW 2 931 MEM 2 958 OKC 2 941 MEM 2 922 MCI 2 952 MEM 2 (AA) 960 VPS 3 930 MEM 3 913 DTW 3 938 MEM 3 909 DFW 3 947 MEM 3 953 MSY 4 924 MEM 4 908 DFW 4 935 MEM 4 917 LIT 4 939 MEM 4 906 ATL 4 949 MEM 4 (AA) 956 OKC 5 925 MEM 5 903 ATL 5 932 MEM 5 962 VPS 5 942 MEM 5 955 MSY 5 948 MEM 5 916 LIT 6 926 MEM 6 912 DTW 6 904 ATL 6 940 MEM 6 959 OKC 6 951 MEM 6 (AA) 919 MCI 7 928 MEM 7 957 OKC 7 937 MEM 7 914 DTW 7 946 MEM 7 907 DFW 7

NA008 923 MEM 1 910 DTW 1 931 MEM 1 958 OKC 1 941 MEM 1 922 MCI 1 952 MEM 1 (AA) 960 VPS 2 930 MEM 2 913 DTW 2 938 MEM 2 909 DFW 2 947 MEM 2 953 MSY 3 924 MEM 3 908 DFW 3 935 MEM 3 917 LIT 3 939 MEM 3 906 ATL 3 949 MEM 3 (AA) 956 OKC 4 925 MEM 4 903 ATL 4 932 MEM 4 962 VPS 4 942 MEM 4 955 MSY 4 948 MEM 4 916 LIT 5 926 MEM 5 912 DTW 5 904 ATL 5 940 MEM 5 959 OKC 5 951 MEM 5 (AA) 919 MCI 6 928 MEM 6 957 OKC 6 937 MEM 6 914 DTW 6 946 MEM 6 907 DFW 6 929 MEM 7 961 VPS 7 933 MEM 7 921 MCI 7 944 MEM 7 918 LIT 7

NA009 960 VPS 1 930 MEM 1 913 DTW 1 938 MEM 1 909 DFW 1 947 MEM 1 953 MSY 2 924 MEM 2 908 DFW 2 935 MEM 2 917 LIT 2 939 MEM 2 906 ATL 2 949 MEM 2 (AA) 956 OKC 3 925 MEM 3 903 ATL 3 932 MEM 3 962 VPS 3 942 MEM 3 955 MSY 3 948 MEM 3 916 LIT 4 926 MEM 4 912 DTW 4 904 ATL 4 940 MEM 4 959 OKC 4 951 MEM 4 (AA) 919 MCI 5 928 MEM 5 957 OKC 5 937 MEM 5 914 DTW 5 946 MEM 5 907 DFW 5 929 MEM 6 961 VPS 6 933 MEM 6 921 MCI 6 944 MEM 6 918 LIT 6 950 MEM 7 902 ATL 7 911 DTW 7 936 MEM 7 905 ATL 7 945 MEM 7 915 DTW 7 (AA)

---

**ruleConfig.txt file**

| CONFIG | COMMON | MAINT | MAXDUTY | 8IN24 | 30IN7 |
|---|---|---|---|---|---|
| SDE | X | O | O | O | O |
| DGE | O | O | X | O | O |
| EAE | O | O | O | O | O |
| EIE | O | O | O | O | O |
| LAE | O | O | O | O | O |
| LIE | O | O | O | O | O |
| ERE | O | O | X | X | X |
| LGE | O | O | X | X | X |
| AAE | X | O | O | O | O |
| TDE | O | O | O | O | O |
| AGE | X | X | X | X | X |
| SRE | X | O | O | O | O |

---

**Servicerate.txt file**

| | | | | | |
|---|---|---|---|---|---|
| 1 | 0000 | NORMAL | DFW | 28 | 30 |
| 1 | 0000 | NORMAL | LIT | 28 | 30 |
| 1 | 0000 | NORMAL | MCI | 28 | 30 |
| 1 | 0000 | NORMAL | MSY | 28 | 30 |
| 1 | 0000 | NORMAL | OKC | 28 | 30 |
| 1 | 0000 | NORMAL | VPS | 28 | 30 |
| 1 | 0000 | NORMAL | DTW | 28 | 30 |
| 1 | 0000 | NORMAL | ATL | 28 | 30 |
| 1 | 0000 | NORMAL | MEM | 28 | 30 |
| 3 | 0700 | BUSY | MEM | 15 | 20 |
| 3 | 0800 | NORMAL | MEM | 15 | 20 |

---

**StationTypeCurfew.txt file**

| | | | |
|---|---|---|---|
| ATL | S | 0600 | 2300 |
| DFW | S | 0600 | 2300 |
| DTW | S | 0600 | 2300 |
| LIT | S | 0600 | 2300 |
| MCI | S | 0600 | 2300 |
| MEM | H | 0000 | 2359 |
| MSY | S | 0600 | 2300 |

| | | | |
|---|---|---|---|
| OKC | S | 0600 | 2300 |
| VPS | S | 0600 | 2300 |

# REFERENCES


Abara J., Applying Inter Linear Programming to the Fleet Assignment Problem, Interfaces, 19, pp20-28. 1989.

Ageeva Y. Approaches to Incorporating Robustness into Airline Scheduling, M Eng. Dissertation, Massachusetts Institute of Technology. 2000.

Arguello M.F., Bard J.F. and Yu G., A GRASP for aircraft routing in response to groundings and delays, Journal of Combinatorial Optimzation, 5 pp211-228. 1997.

Barnhart C. Planning and Control of Transportation Systems: Robust Airline Planning, technical report, Massachusetts Institute of Technology. 2001.

Barnhart C., Boland N.L., Clarke L.W., Johnson E.L., Nemhauser G.L. and Shenoi R.G. Flight String Models for Aircraft Fleeting and Routing, Transportation Science, 32(3), pp208-220. 1998.

Bureau of Transportation Statistics. *2001 Airline On-time Performance*. 2002.

Clarke L., Johnson E.L., Nemhauser G. and Zhu Z. The aircraft rotation problem, Transportation Research Part A: Policy and Practice, 30(1), pp51. 1996a.

Clarke L.W., Hane C.A., Johnson E.L. and Nemhauser G.L. Maintenance and Crew Consdierations in Fleet Assignment, Transportation Science, 30(3), p249-260. 1996b.

Coelle Coelle Carlos A. A Short Tutorial on Evolujtionary Multiobjective Optimzation, First International Conference, EMO 2001, Zurich, Switzerland, March 2001, Proceedings, p21-40. 2001.

Deb K. Multi-objective Optimzation using Evolutionary Algorithms, Wiley. 2001.

Deb K. and Goldberg D.E. An Investigation of Niche and Species Formation in Genetic Function Optimization. Genetic Algorithms: Proceedings of the third International Conference. Morgan Kaufmann, pp42-50. 1989.

Dias A.H.F. and Vasconcelos J.A. Multiobjective Genetic Algorithms Applied to solve Optimization problems, IEEE Transactions on Magnetics, 38(2), pp1133-1136. 2002.

Ehirgott M. and Ryan D.M. Constructing Robust Crew Schedules with Bicriteria Optimzation, Jornal of Multi-Criteria Decision Analysis, 11, pp(1-12). 2002.

Filar J.A. Manyem P. and White K. How Airlines and Airports Recover from Schedule Perturbations: A Survey, Annals of Operations Research, 108, pp315-333. 2001.

Fonseca C.M. and Fleming P.J. Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization, Genetic Algorithms: Proceedings of the fifth International Conference. Morgan Kaufmann, San Mateo, CA pp416-423. 1993.

Fonseca C.M. and Fleming P.J. Multiobjective Optimization and Multiple Constraint Handling with Evolutionary Algorithms – Part I: Unified Formulation, IEEE Transactions on Systems, Man and Cybernetics – Part A: Systems and Human, 28 (1) pp26-37. 1998.

Gen M. and Cheng R.W. Genetic algorithms and engineering optimization, Wiley, New York. 2000.

Goldberg D.E. Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, Reading, MA. 1989.

Gopalan R. and Talluri K.T. Mathematical models in airline schedule planning: A Survey, Annals of Operations Research, 76, pp155-185. 1998.

Graves G.W., McBride R.D., Gershkoff I., Anderson D. and Mahidhara D. Flight Crew Scheduling, Management Science, 39, 736-745. 1993.

Grosche T., Heinzl A. and Rothlauf F. A Conceptual Approach for Simultaneous Flight Schedule Construction with Genetic Algorithms, Applications of evolutionary computing : EvoWorkshops 2001 : EvoCOP, EvoFlight, EvoIASP, EvoLearn, and EvoSTIM, Como, Italy, April 2001 : proceedings. 2001.

Haeme R.A., Huttinger J.L. and Shore R.W. Airline performance modelling to support schedule development: An application case study, Proceedings of the 1998 Winter Simulation Conference, pp800-805. 1998.

Hane C.A., Barnhart C. amd Sigismindi G. The fleet assignment problem: solving a large scale integer program, The Mathematical Programming Society, 70, pp211-231.1995.

Hane C.A., Barnhart C., Johnson E.L., Marsten R.E., Nemhauser G.L. and Sigismondi G. The Fleet Assignment Problem: Solving A Large-Scale Integer Program, Mathematical Programming, 70, pp211-232. 1995.

Hoffman K.L., and Padberg M. Solving Airline Crew Scheduling Problems by Branch-and-Cut, Management Science 29, pp657-682. 1993.

Horn J., Nafpliotis N. and Goldberg D.E. A Niched Pareto Genetic Algorithm for Multiobjective Optimization. Proceedings of the First IEEE Conference in Evolutionary Computation, IEEE World Congress on Computational Intelligence, 1, pp82-87. 1994.

Horn J., Nafpliotis N. and Goldberg D.E. A Niched Pareto Genetic Algorithm for Multiobjective Optimization, Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence, 1, pp82-87. 1994.

Ishibuchi H. and Murata T. Multi-objective Genetic Local Search Algorithm and its Application to Flowshop Scheduling, IEEE Transcations on Systems, Man and Cybernetics Part C, 28(3), pp392-403. 1998.

Jaszkiewicz Z. Genetic Local Search for multi-objective combinatorial optimization, EJOR, 137, pp50-71. 2002.

Jones D.F., Mirrazavi S.K. and Tamiz M. Multi-objective meta-heuristics: An Overview of the current state-of-the-art, EJOR, 137, pp1-9. 2002

Langerman J.J. and Ehlers E.M. Agent-based airline scheduling, Computers ind. Engng, 33(3-4), p849-852. 1997.

Davis L. Handbook of Genetic Algorithms, Van Nostrand Reinhold, New York. 1991.

Lee L.H., Huang H.C., Lee C., Chew E.P., Wikrom J, Yong Y.Y., Liang Z., Leong C.H., Tan Y.P., Namburi K., Johnson E.L. and Banks J. Discrete Event Simulation Model for Airline Operations: SIMAIR, submitted to the Winter Simulation Conference. 2003.

Lettovsky L. Airline Operations Recovery: An Optimization Approach, Ph D Dissertation, Georgia Institute of Technology. 1997.

Lettovsky L., Johnson E.L. and Nemhauser G.L. Airline Crew Recovery, Transportation Science, 34(4). 2000.

Li D., Yan J.B. and Biswal M.P. Quantitative parametric connections between methods for generating non-inferior solutions in multiobjective optimization, EJOR, 117, pp84-99. 1999.

Listes O. and Dekker R. A scenario aggregation based approach for determining a robust airline fleet composition, Econometric Institute Reports EI 2002-17 Erasmus University Rotterdam. 2002.

Mederer M. and Frank M. Increasing Robustness of Flight Schedules through Stochastic Modelling of Planning Parameters, presented at the AGIFORS 2002, 42 Annual Symposium, Honolulu. 2002.

Michalewicz Z., Dasgupta D. Evolutionary algorithms in engineering applications, Berlin: New York: Springer. 1997.

Miettinen K. Some Methods for nonlinear Multi-objective Optimizaiton, First International Conference, EMO 2001, Zurich, Switzerland, March 2001. Proceedings, pp1-20. 2001.

Murata T. and Ishibuchi H. MOGA: Multi-Objective Genetic Algorithms, Proceedings of the 2nd IEEE International Conference of Evolutionary Computation, pp289-294. 1995.

Pareto V. Cours D'Economic Politique, volume I and II, F Rouge, Lausanne. 1896.

Rexing B., Barnhart C., Kniker T., Jarrah A. and Krishnamurthy N. Airline Fleet Assignment with Time Windows, Transportation Science, 34(1), pp1-20. 2000.

Rosenberg R.S. Simulation of genetic populations with biochemical properties, PhD thesis, University of Michigan, Ann Harbor Michgan. 1967.

Rosenberger J.M. Topics in Airline Operations, PhD Dissertation, Georgia Institute of Technology. 2001a.

Rosenberger J.M., Johnson E.L. and Nemhauser G.L. A Robust Fleet Assignment Model with Hub Isolation and Short Cycles, Technical report, Georgia Institute of Technology. 2001b.

Rosenberger J.M., Johnson E.L. and Nemhauser G.L. Rerouting Aircraft for Airline Recovery, Technical report, Georgia Institute of Technology. 2001c

Rosenberger J.M., Schaefer A.J., Goldsman D., Johnson E.L. Kleywegt A.J. and Nemhauser G.L. A Stochastic model of Airline Operations, Transportation Science, 36, pp 357-377, 2002.

Schaefer A.J., Johnson E.L., Kleywegt A.J. and Nemhauser G.L. Airline Crew Scheduling Under Uncertainty, Technical report, Georgia Institute of Technilogy. 2001.

Schaffer J.D. Multi Objective Optimization with Vector Evaluated Genetic Algorithms. Genetic Algorithms and their Applications: Proceedings of the First International Conference in Genetic Algorithms, Lawrence Erlbaum, pp 93-100. 1985.

Srinivas N. and Deb K. Multi-objective Optimization Using Nondominated Sorting in Genetic Algorithms. Evolutionary Computation, 2(3), pp 221-248. 1994

Stojkovic M., Soumis F. and Desrosiers J. The Operational Airline Crew Rescheduling Problem, Transportation Science, 32, pp232-245. 1998.

Subramanian R., Scheff Jr R.P. and Quillianan J.D. Coldstart: Fleet Assignment at Delta Airlines, Interfaces, 24, pp104-120. 1994.

Teodorovic D. and Stojkovic G. Model for Operational Daily Airline Scheduling, Transoprtation Planning and Technology, 14, pp273-285. 1990.

Teodorovic D. and Stojkovic G. Model to Reduce Airline Schedule Disturbances, Journal of Transportation Engineering, 121, pp324-331. 1995.

Teodorovic D., and Guberinic S. Optimal dispatching strategy on an airline network after a schedule perturbation. European Journal of Operational Research, 15, pp178-182. 1984.

The Atlanta Journal-Constitution. Flight-delay problems down in ’02, reports FAA, http://www.accessatlanta.com/ajc/business/delta/1202/06faadelay.html. 2002.

Thengvall B.G., Yu G. and Bard J.F. Balancing use preferences for aircraft schedule recovery during irregular operations. IIE Transactions 32, pp181-193. 2000.

Thengvall B.G., Yu G., and Bard J.F. Multiple Fleet Aircraft Schedule Recovery Following Hub Closures, Transportation Research Part A, pp289-308. 2001.

Vance P.H., Atamturk A., Barnhart C., Gelman E., Johnson E.L. and Krishna A., Madidhara D., Nemhauser G.L. and Rebello R. A heuristic branch-and-price approach for the airline crew pairing problem, Technical Report TLI/LEC-97-06, Georgia Institute of Technology, Alanta, GA. 1997.

Vance P.H., Barnhart C., Johnson E.L. and Nemhauser G.L. Airline Crew Scheduling: A New Formation and Decomposition Algorithm, Operation Research, 45, pp188-200. 1997.

Wells A.T. Air Transportation: A Management Perspective, Belmont, CA : Wadsworth Publishing, Fourth Edition. 1999.

Wienke D., Lucasius C. and Kateman G. Multicriteria target vector optimization of analytical procedure using a genetic algorithm, Part I Theory, numerical simulations and application to atomic emission spectroscopy, Analytica Chimica Acta, 265, pp211-225. 1992

Wu C. and Caves R.E. Towards the optimisation of the schedule reliability of aircraft rotations, Journal of Air Transport Management, 8, pp419-426. 2002.

Yan S. and Yang D. A decision support framework for handling schedule perturbations, Transportation Research. Part b: Methodology, 30(6), pp405-419. 1997.

Yen J. and Birge J. A Stochastic Programming Approach to The Airline Crew Scheduling Problem, The 35th Annual Conference of the Operational Research Society of New Zealand. 2000.

Yu G. Operations Research in the Airline Industry, Boston : Kluwer Academic Publishers. 1998.

Zalzala A.M.S. and Fleming P.J. Genetic algorithms in engineering systems, London : Institute of Electrical Engineers. 1997.